
MICRO:BIT

造物粒子套件

入门教程



DFROBOT
DRIVE THE FUTURE

目录

第一话：MakeCode 与 micro:bit	5
MakeCode 平台介绍	5
Micro:bit 简介	5
初次使用 micro:bit	6
STEP1: 打开 makecode 在线编程平台	7
STEP2: 连接 micro:bit 至电脑	7
STEP3: 新建项目	8
STEP4: 下载程序至 micro:bit	10
第二话：是什么让我们的机器“活”过来的?.....	13
交互设备	13
输入单元——传感器	13
控制单元——Micro:bit 主板	13
输出单元——执行机构.....	13
程序和硬件之间的关系	13
Micro:bit 扩展板.....	14
第三话：小试牛刀.....	15
牛刀小试	15
项目一 神秘的 micro:bit	15
所需元件	15
硬件连接	16
编写程序	16
能力提升	19
项目二 闪烁的 LED	20
所需物品	20
硬件连接	21
编写程序	21
能力提升	23
项目三 呼吸灯	24
所需元件	24
硬件连接	25
编写程序	25

能力提升	29
项目四 变速风扇.....	30
所需元件	30
硬件连接	31
编写程序	32
第四话：如虎添翼.....	36
如虎添翼	36
项目一 电子蜡烛.....	36
所需元件	36
硬件连接	37
编写程序	37
项目二 自动门.....	41
所需元件	41
硬件连接	42
编写程序	42
项目三 音乐盒.....	47
所需元件	47
硬件连接	48
编写程序	48
项目四 炫彩灯带.....	52
所需元件	52
硬件连接	53
编写程序	53
第五话：大展拳脚.....	60
大展拳脚	60
项目一 电子稳定器.....	60
所需元件	60
硬件连接	61
编写程序	61
能力提升	63
项目二 DJ 演奏台.....	64
所需元件	64
硬件连接	65
编写程序	65

能力提升	67
项目三 可移动门铃	68
所需元件	68
硬件连接	69
编写程序	70
能力提升	72
项目四 密室逃脱.....	73
所需元件	73
硬件连接	73
编写程序	74
能力提升	79

*点击页面左上角 DF 创客社区，有意外惊喜哦！

第一话：MakeCode 与 micro:bit

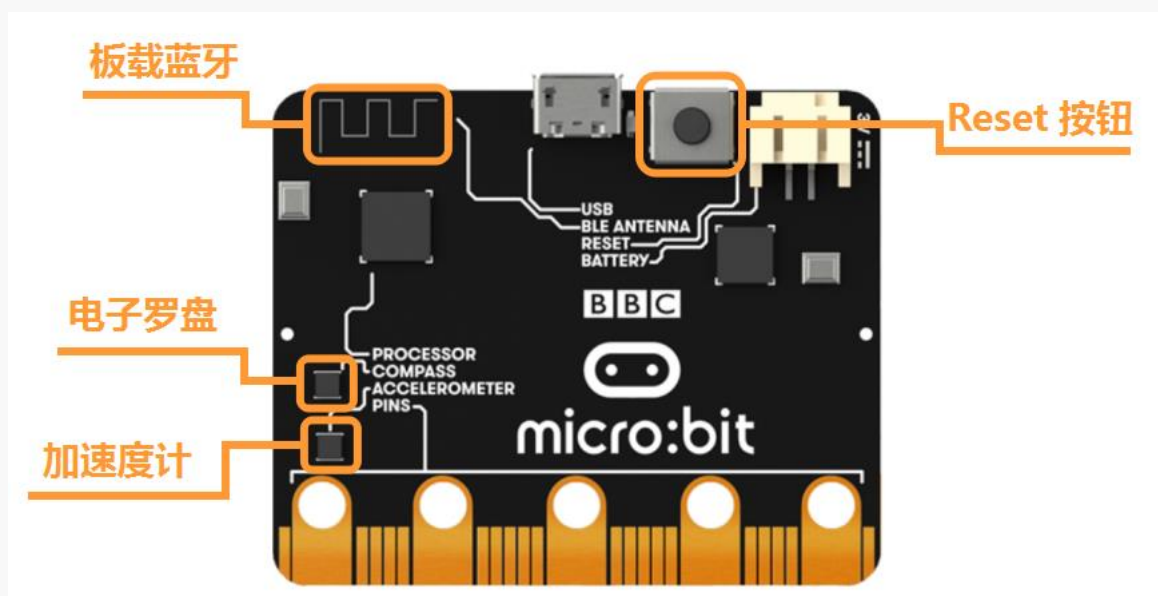
MakeCode 平台介绍

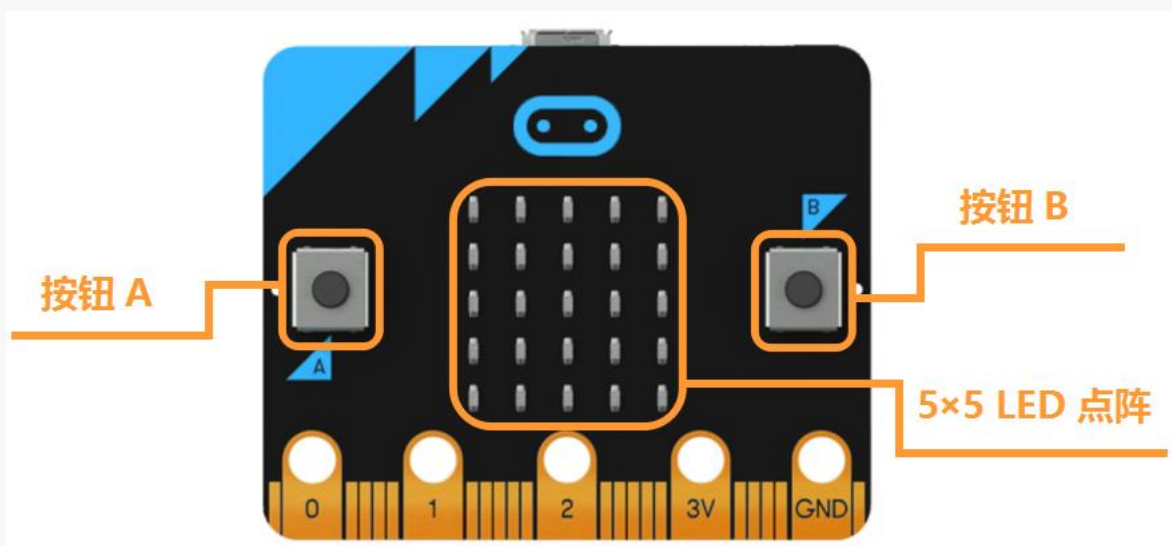
MakeCode for micro:bit 是 micro:bit 官网上使用最为广泛的图形化编程环境，它基于微软开源项目 MakeCode 开发的图形化编程环境。

Micro:bit 简介

Micro:bit 是一款由英国 BBC 设计的 ARM 架构的单片机，板载蓝牙，加速度计，电子罗盘，三个按钮，5×5LED 点阵，主要用于青少年的编程教育。

利用 micro:bit 可以实现任何酷炫的小发明，无论是机器人还是乐器，没有想不到。micro:bit 拥有一系列新颖的功能，例如 25 个可显示消息的红色 LED 灯；有两个可编程按钮，可以用于控制游戏操作或者暂停/播放一首音乐。micro:bit 可以检测动作并且告知使用者动作进行的方向，同时它也可以通过低功耗蓝牙模块与其它设备或因特网互联。





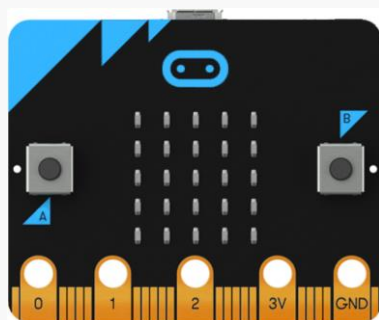
需要注意的是，micro:bit 还自带光照度传感器和温度传感器等常见的感应装置，这意味着它自己本身也可以制作很多生活中常见的智能化产品。

初次使用 micro:bit

如果你是第一次接触 micro:bit，同样可以通过以下零基础教程学习如何使用 makecode 在线编程平台，以及如何为 micro:bit 编写程序。

开始之前，请确认你手边有如下物品，除此之外，你还需要一台运行 Windows/Mac OS/Linux 操作系统并且有网络连接的电脑。

所需元件：



Micro:bit 主板



USB A to B 连接线

STEP1: 打开 makecode 在线编程平台

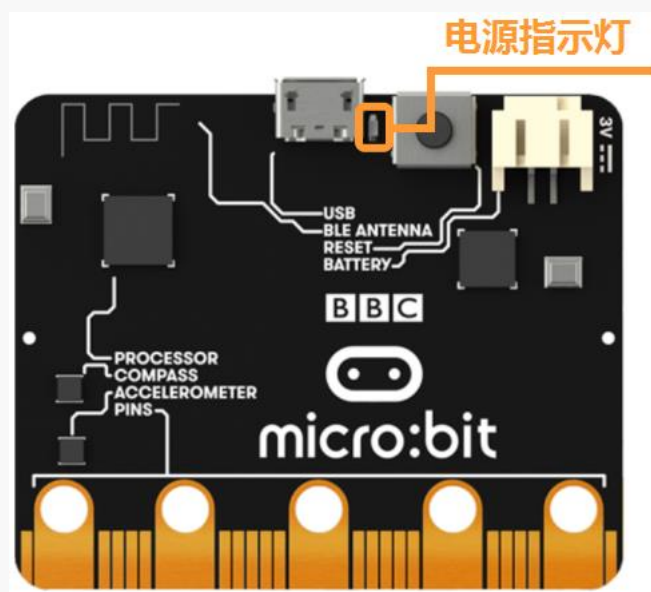
以下的步骤说明基于 Windows 操作系统，如果你使用的是其他操作系统，可以将其作为参考。

首先，你需要登录网页：<http://microbit.dfrobot.com.cn/index.html>

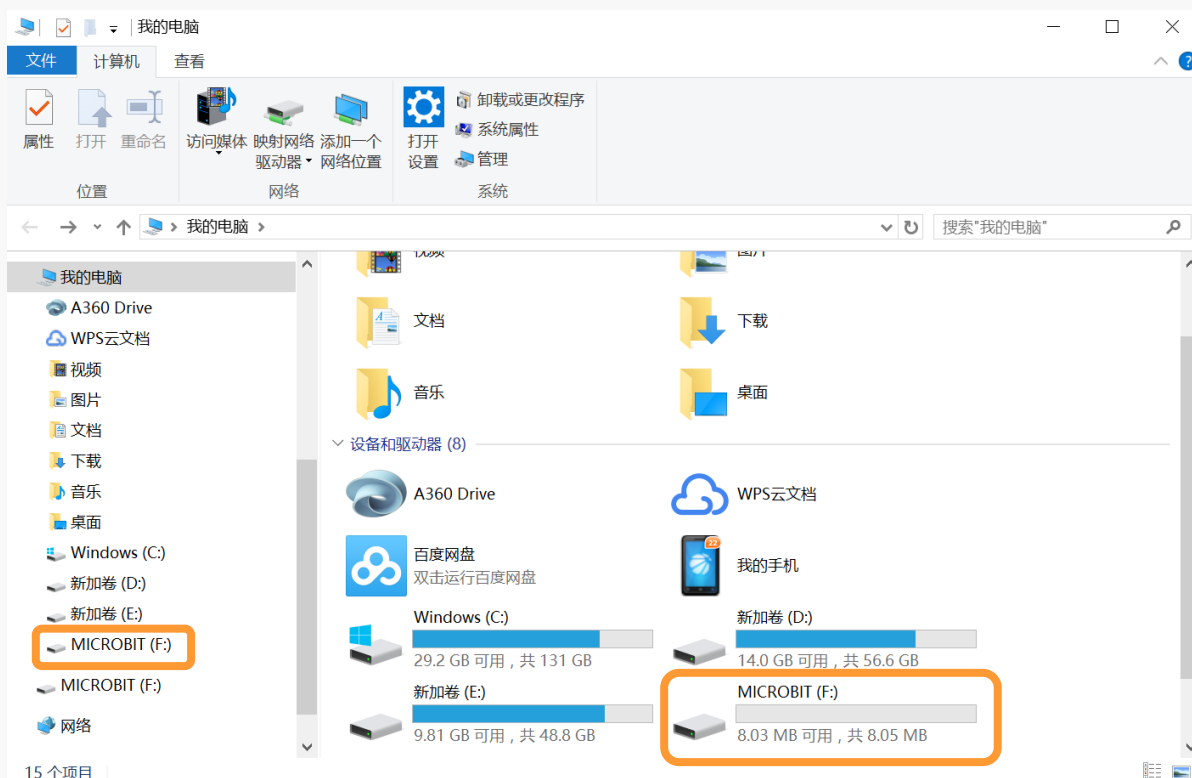


STEP2: 连接 micro:bit 至电脑

将 micro:bit 主板通过 USB A to B 数据线连接至电脑。正确连接时 micro:bit 主板背面的电源指示灯会亮起。



在编程之前，我们需要确保主板被电脑识别。 打开“我的电脑”。接上 micro:bit 主板的就会在磁盘列表中显示磁盘“MICROBIT”。



STEP3: 新建项目

现在让我们仔细看下 makecode 在线编程界面。

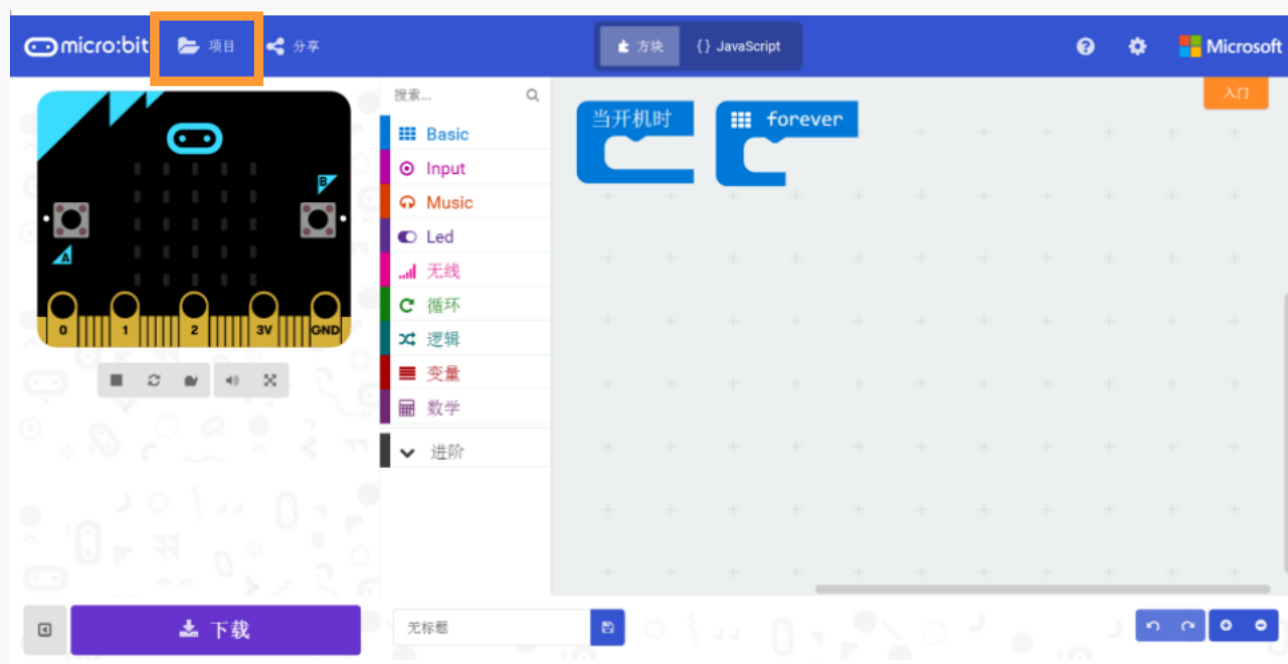


模拟窗口：模拟 micro:bit 工作的状态，在编程过程中，可以通过观察模拟窗口，观看初步的运行结果。

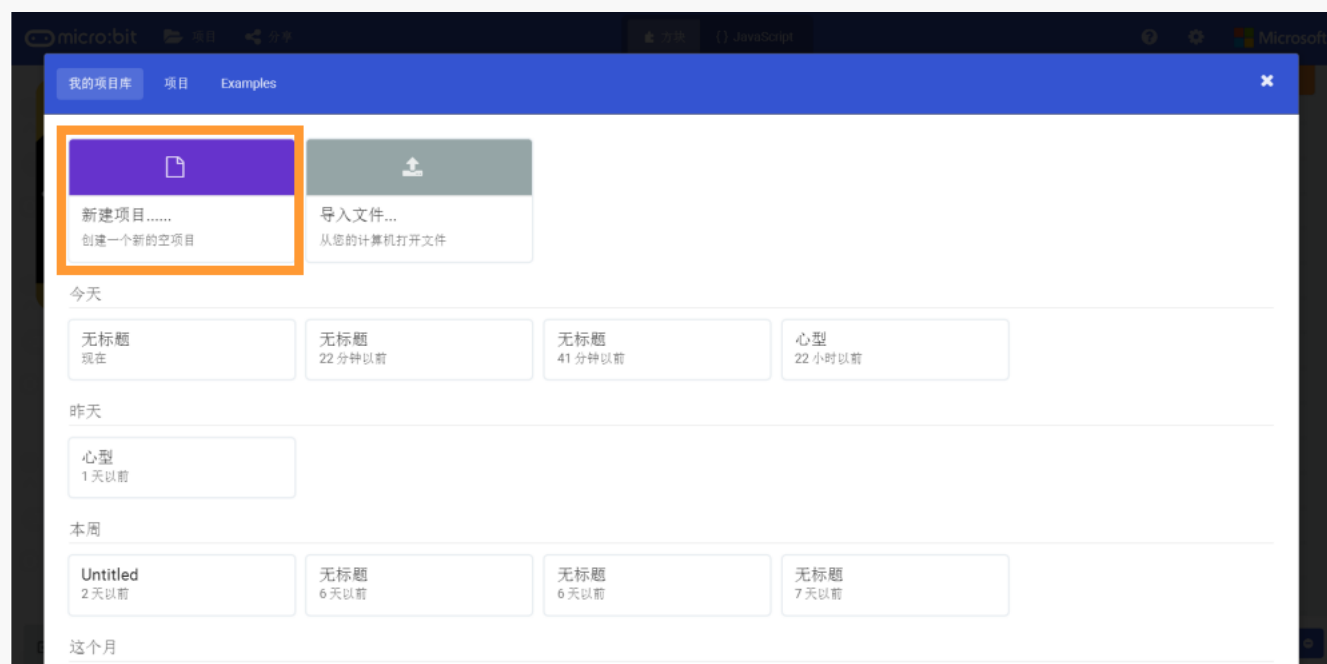
指令区：拖动指令区的指令可以对 micro:bit 进行编程控制。

脚本区：程序的编写区域，拖拽指令区的指令在此编写程序。

单击模拟窗口正上方的“项目”。



在跳转出来的页面单击“新建项目”即可。



STEP4: 下载程序至 micro:bit

程序编写完成之后要下载到 micro:bit 主板里面，点击指令区正下方的“无标题”框，给程序命名。



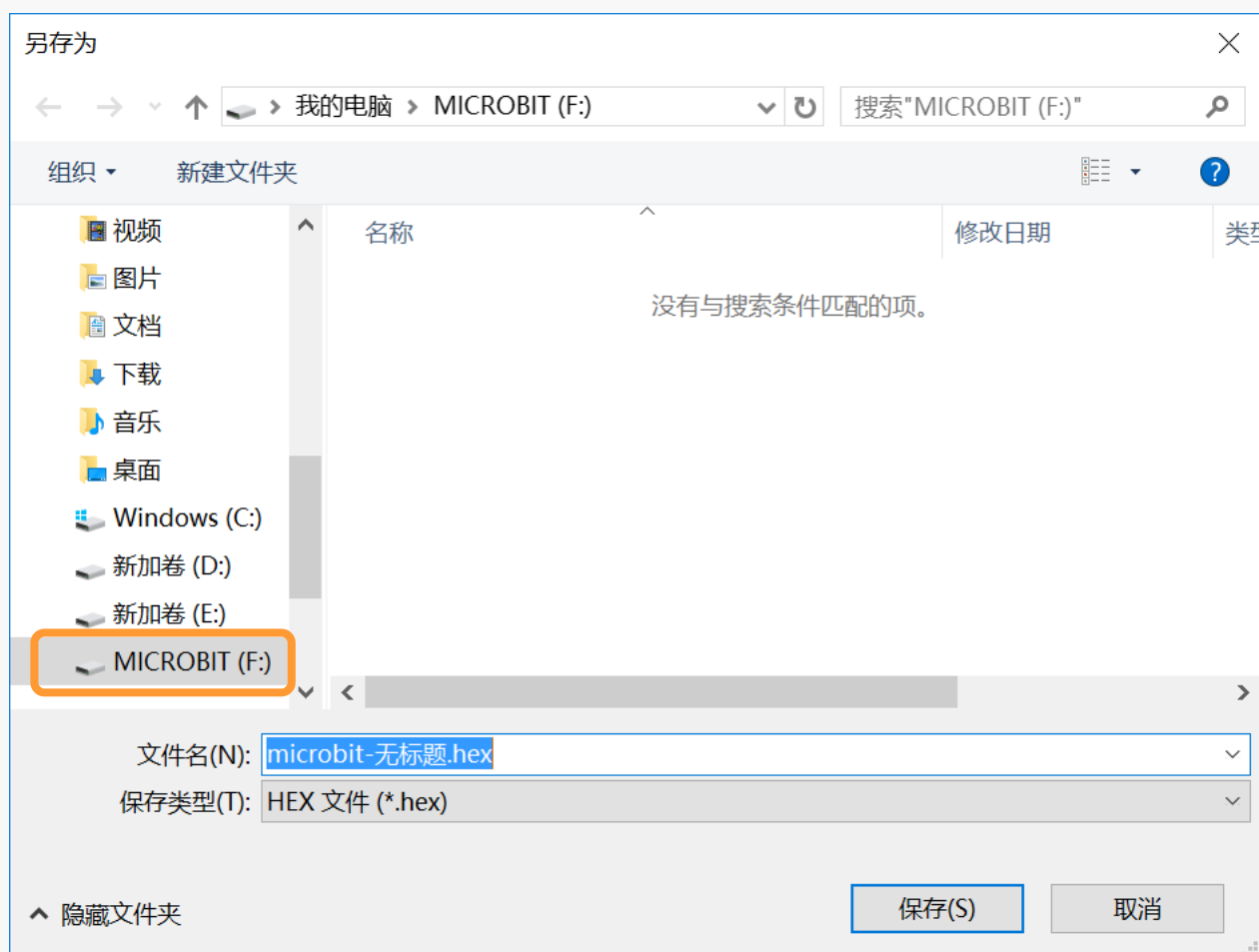
在下载之前，你可以在模拟窗口看到程序运行之后的模拟动作，通过模拟窗口下方的按钮可以控制模拟器。



点击模拟窗口正下方的“下载”，弹出的对话框下选择“另存为”。



在弹出的对话框中，选择把.hex 文件保存到“MICROBIT 磁盘”，再点击“保存”。



下载过程中 micro:bit 主板背面的电源信号灯会闪烁，当下载完成后电源信号灯停止闪烁，保持长亮。

以上就是使用 micro:bit 的一些基本的方法。如果在使用中存在任何疑问或者建议，欢迎访问我们的论坛联系我们。

论坛链接: <http://www.dfrobot.com.cn/community/forum.php>

记得来逛 DFRobot 的社区看看更多的教程和精彩的项目哦。我们同样希望你能够把你自己的项目或者想法发在论坛上分享。欢迎成为我们的一员！

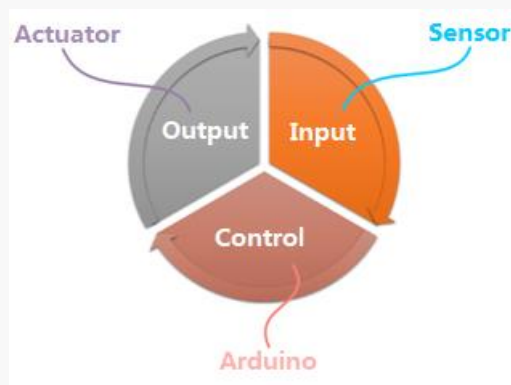
第二话：是什么让我们的机器“活”过来的？

交互设备

在接下来的几个章节中，我们将建立一些交互套件项目，从点亮一个 LED 灯开始，到显示炫酷的彩虹灯带和音乐设置。这些套件同样可以被分类为“交互式设备”，为了帮助你更好地理解他们的机械原理，我们需要首先弄明白它们的构造。

最简单的交互设备也要由 3 个部分组成：

- 用于命令接受或数据收集的输入单元
- 用于数据处理或信号处理的控制单元
- 用于发送数据或执行动作的输出单元



那么这些单元都意味着什么呢？举我们自身为例，我们通过眼睛、耳朵、鼻子和皮肤收集光线、声音、味道和力量形式的信息，这些信息会进入我们的大脑来决定需要采取什么反应。最终我们根据这些信息采取实际行动来改变周身物理环境。具体来说，例如你的朋友对你说“你好”，你回应说“你好”，你的耳朵扮演着输入元件、大脑扮演控制模块，嘴巴扮演输出单元这样的角色。

同样，对 micro:bit 项目来说，我们使用多种传感器作为输入单元，micro:bit 主板作为控制单元，执行机构作为输出单元。

输入单元——传感器

传感器（又被称为变换器）是一种可以察觉或探测到环境特性（比如光、温度、湿度等）并且将它们转化为信号数据的物理元件。在本教程中，我们将使用按钮、声音传感器和温度传感器等传感器。

控制单元——Micro:bit 主板

Micro:bit 主板将在这扮演控制单元，使用信号引脚建立输入单元和输出单元的联系，并且在计算处理模块中处理数据。

输出单元——执行机构

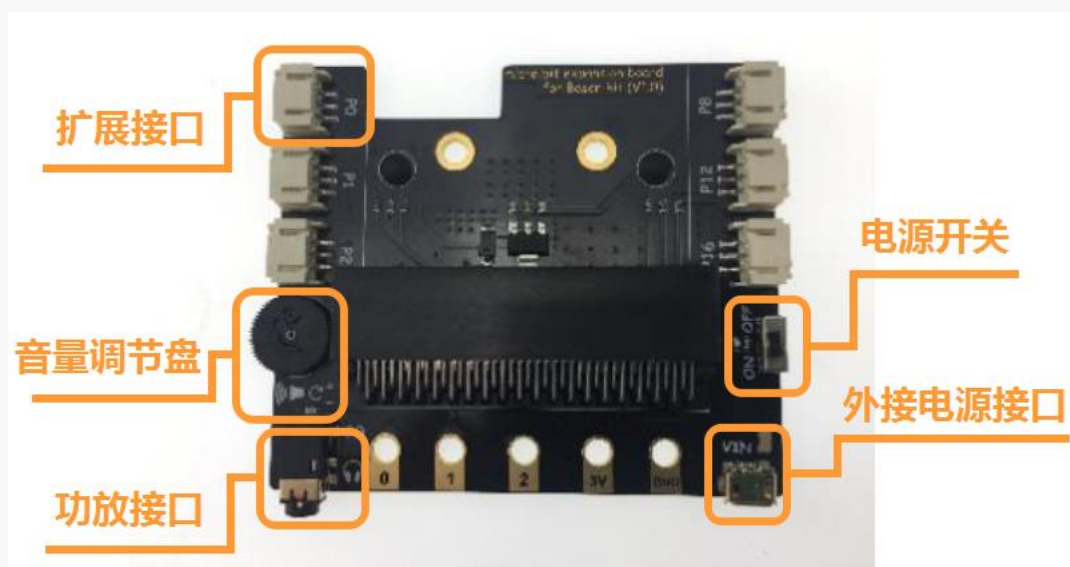
执行机构是负责移动或者控制一个系统或者机械结构的设备。它把电能转化为运动，声音和光。在本教程中，我们使用类似 LED 灯、小风扇和舵机等执行机构。

程序和硬件之间的关系

上面提到的输入单元，控制单元和输出单元都是硬件，类比我们人类，硬件是我们的身体。然而，大脑更加重要，因为它产生想法并控制我们的每一个行为。在这里程序就像我们的大脑一样。对我们人类来说，身体和大脑都是不可或缺的。

Micro:bit 扩展板

Micro:bit 扩展板可以扩展按钮和开关，扩展电源接口和其他模块接口，模块可以直接连接在扩展板上，这使得回路连接更为快速而高效，也使得 micro:bit 功能可以更为强大。



第三话：小试牛刀

牛刀小试

经过前面几个章节的知识介绍，你是不是跃跃欲试了呢？现在，我们就开始 Boson for micro:bit 的神奇之旅吧！

项目一 神秘的 micro:bit

在对 MakeCode 平台和 micro:bit 主板有了基础的了解，并认识了扩展板之后，我们就可以来做第一个项目——制作表情包啦。

我们常常在手机上或者电脑上面会见到一些萌萌的，很可爱的表情包，比如这样的：



我们可不可以自己设计表情包，随时随地给朋友们展示呢？

micro:bit 板子上面有 5*5（25 个）板载的小灯，我们可以通过控制小灯的亮暗，制作小灯亮出不同的造型，展示表情包的效果。

所需元件

1 × micro:bit 主控板



1 × USB A to B 连接线



硬件连接

将 micro:bit 主控板用 USB 连接线连接在电脑上。

编写程序

打开 MakeCode 在线编程界面。

STEP1：新建项目，具体操作可参考第一话“初次使用 micro:bit”的 STEP3。

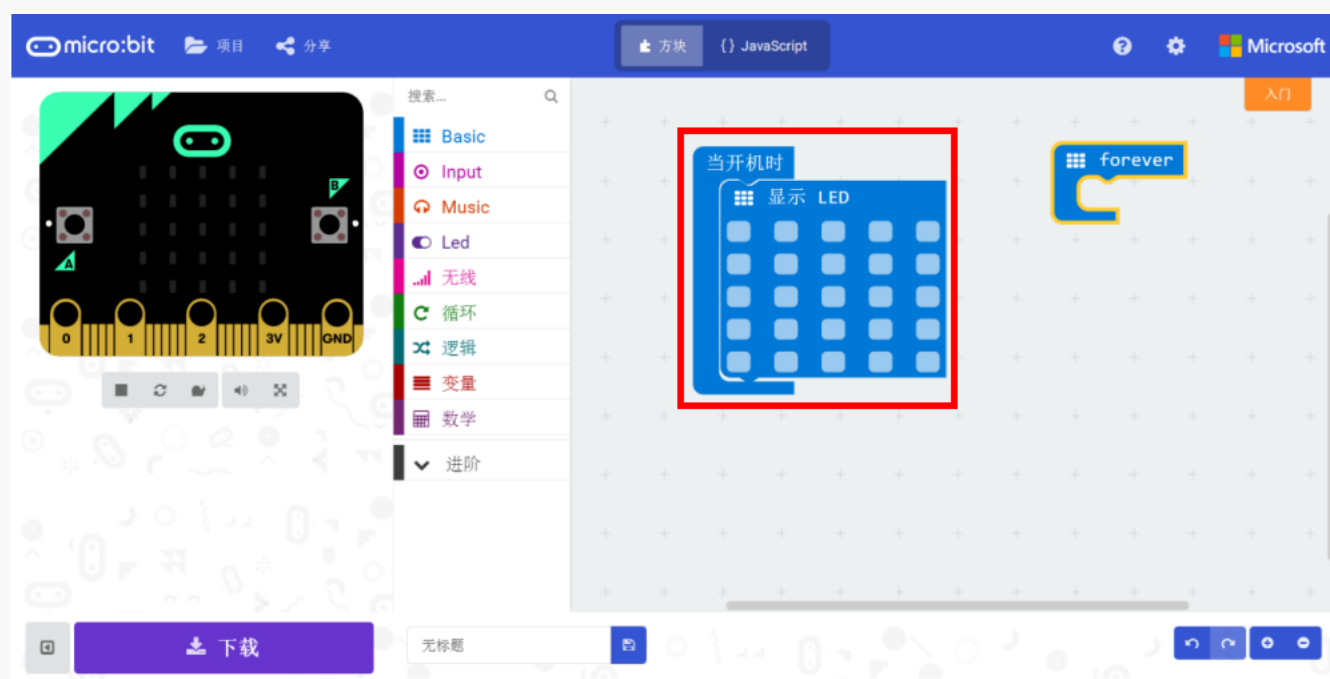
STEP2：单击指令区最上面的“Basic”，在指令区的右侧会出现一排模块，在右侧模块中找到“显示 LED”模块。



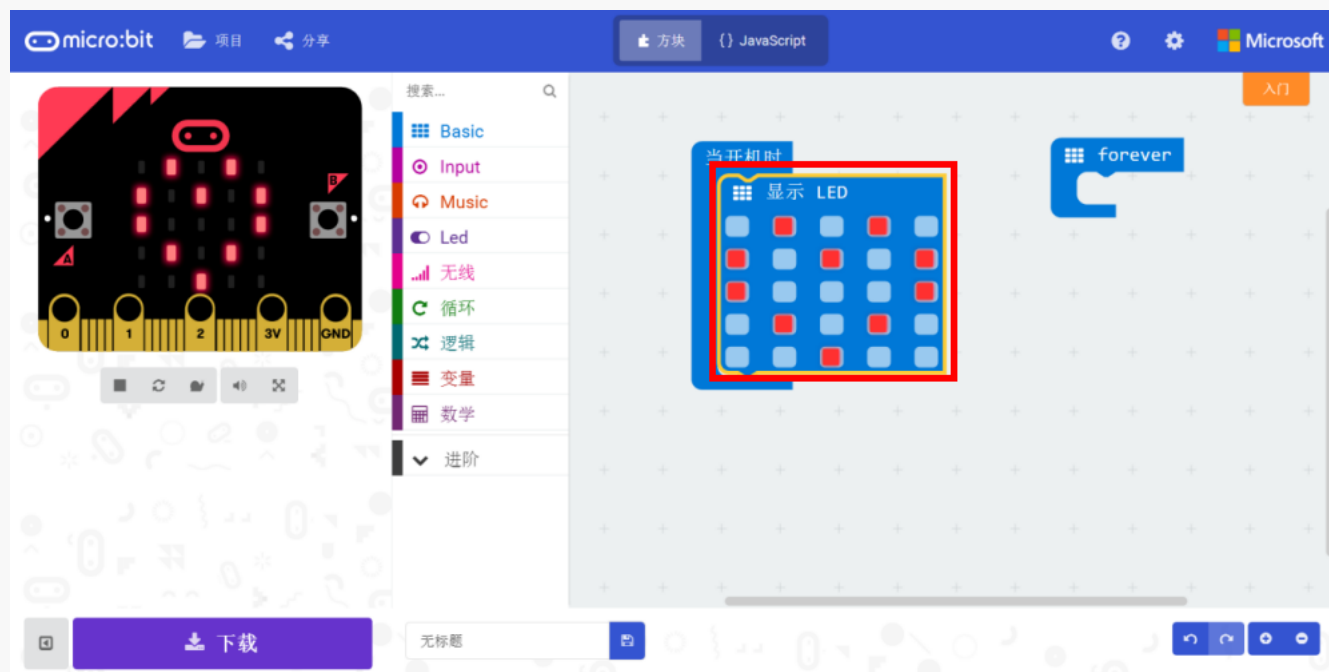
STEP3：打开 MakeCode 在线网站之后，界面上会自动出现“当开机时”模块，或者在“BASIC”右侧模块的最底部可以找到“当开机时”模块，将“当开机时”模块拖至脚本区。



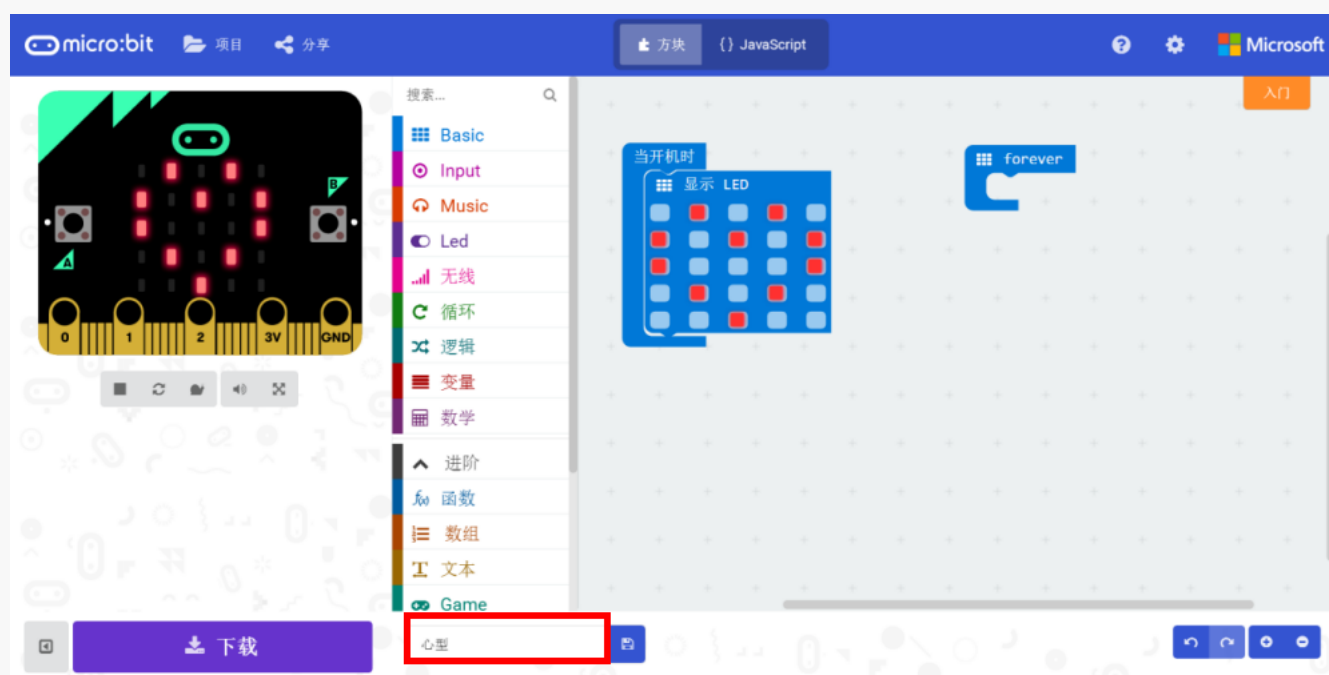
STEP4: 将“显示 LED”模块拖动至脚本区，并放入“当开机时”模块内。



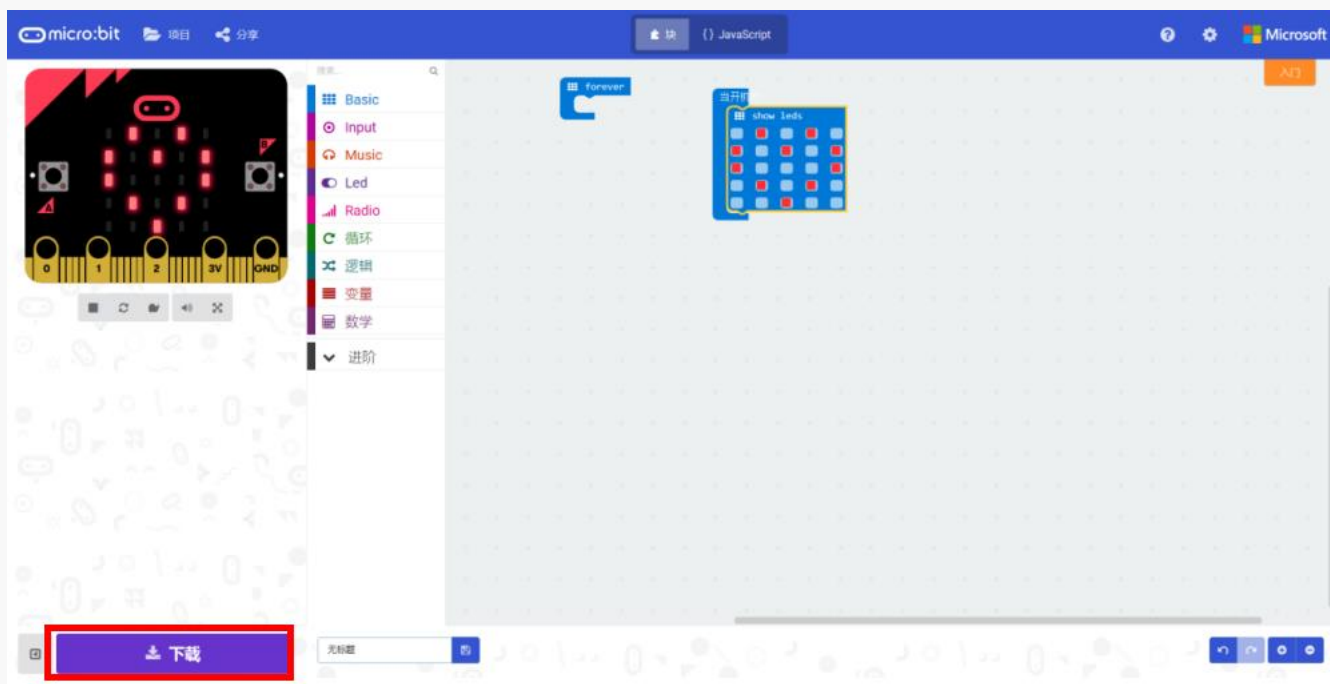
在脚本区可以看到已经拖动的“显示 LED”模块，单击浅蓝色的小方块，可以开始设计表情包，首先，先来绘制一个萌萌的“心型”，可以在左边的模拟窗口可以看到绘制的模拟效果。



STEP5: 在下方红框内，编辑为“心型”（可以自己命名）。



STEP6: 单击“下载”，将文件保存在电脑上“micro:bit”硬盘上，即可完成保存程序，下载至 micro:bit 主板。这样，micro:bit 板子上就会显示“心型”。（具体下载步骤请参考第一话“初次使用 micro:bit”的 STEP4）



能力提升

试着制作可以切换两个表情的表情包。

项目二 闪烁的 LED

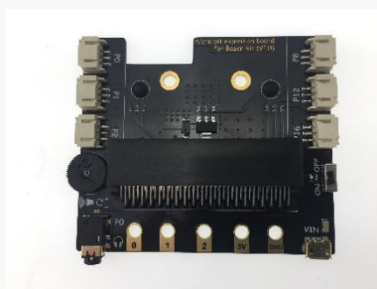
上节课我们学习了点亮板载的小灯，制作了属于自己的表情包，在这一部分，我们将外接一个 LED 小灯，编程使 LED 灯点亮和闪烁。

所需物品

1 × micro:bit 主控板



1 × micro:bit 扩展板



1 × LED 灯模块



1 × USB A to B 连接线



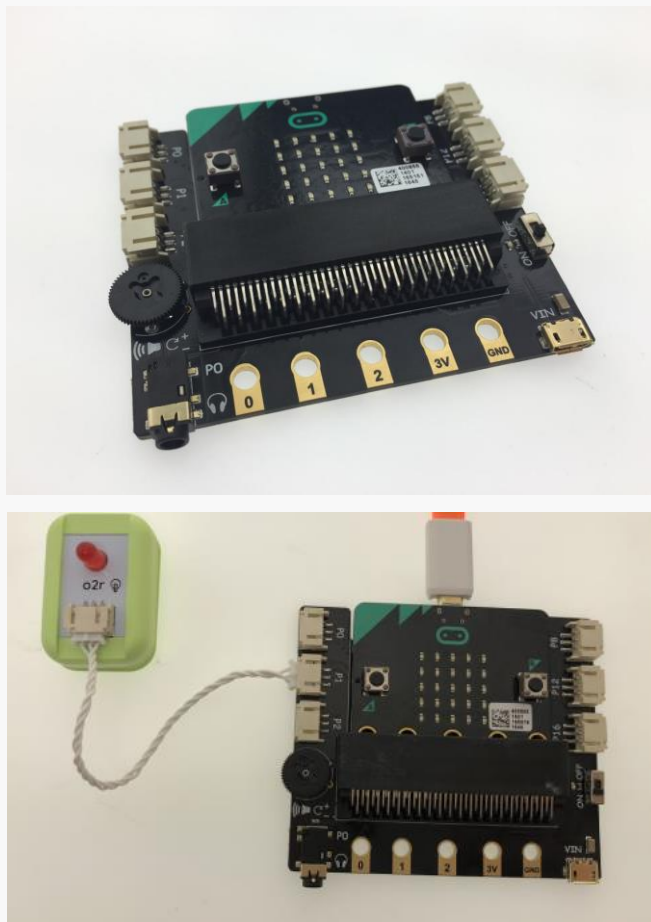
硬件连接

我们需要连接以下部分：

- micro:bit 主控板与 micro:bit 扩展板连接；

注意：接下来要学习的项目中都是将 micro:bit 主控板与扩展板结合在一起使用的，后续这一部分的连接将不会指出。

- LED 灯模块连接到扩展板的 P1 号接口；



编写程序

任务一：点亮 LED 灯

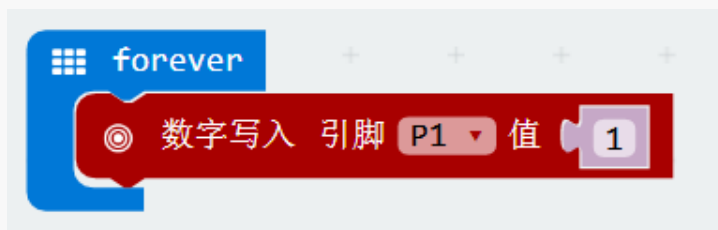
打开 MakeCode 在线编程界面。

STEP1：新建项目，具体操作可参考第一话“初次使用 micro:bit”的 STEP3。

STEP2：硬件连接中运用到了外接 LED 小灯，编程过程中，通过设置 micro:bit 的引脚值（连接 LED 小灯的引脚）来控制小灯的亮暗。控制 LED 小灯呈现“亮”或“暗”两种状态。LED 小灯在这个实验中属于数字输出，单击“进阶”，会出现“Pins”，调用指令的流程为：Pins（进阶指令）----数字读取引脚 p0 值（0）。（拖动后，将 P0 改为 P1）



STEP3: 将其指令拖动入“forever”中，当烧录代码至 micro:bit 小板子时，可以点亮连接在 P1 号引脚的外接 LED 小灯。



任务二：闪烁 LED 灯

在点亮 LED 灯任务的基础上，再让 LED 小灯实现闪烁的功能。

STEP1: 新建项目，具体操作可参考第一话“初次使用 micro:bit”的 STEP3。

STEP2: 设置 micro:bit 的引脚值来控制小灯的亮暗，控制 LED 小灯呈现“亮”或“暗”两种状态。LED 小灯在这个实验中属于数字输出，单击“进阶”，会出现“Pins”，调用指令的流程为：Pins（进阶指令）---数字读取引脚 p0 值（0）。（拖动后，将 P0 改为 P1）引脚值只有 0 和 1 两种情况，当值为 1 时，表示小灯被点亮；当值为 0 时，表示小灯呈现熄灭状态。



STEP3: 还需要让外接 LED 小灯保持亮的状态 1 秒，保持熄灭的状态 1 秒，呈现亮 1 秒，暗 1 秒相交替的状态，需要调用指令的流程为：Pins（进阶指令）---暂停（ms）（100）。“暂停”指令可以通过改变后面的参数，控制 LED 小灯呈现亮或灭的状态保持一定时间。



STEP4: 将“暂停”指令拖入“forever”中，调整数字后，最终代码见下图。表示：P1 引脚为高电平（点亮状态）保持 1000ms，P2 引脚为低电平（熄灭状态）保持 1000ms，循环呈现闪烁效果。



能力提升

通过查找资料，了解一下 SOS 紧急救援信号灯的原理，与同学分享，尝试制作 SOS 信号灯吧！

项目三 呼吸灯

上节课我们学习了点亮外接 LED 灯的编程方法，这节课我们要学习使用按钮控制小灯的亮灭，还要学习使用旋钮制作呼吸灯的方法。

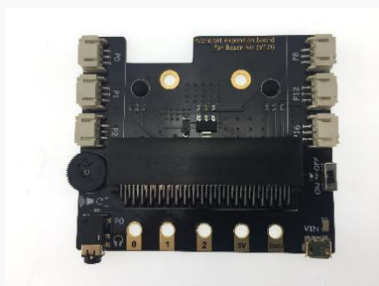
其实呼吸灯是我们日常生活中很常见的提示信号，例如手机来信息时会闪烁的信号灯，现在我们就来认识一下它吧！

所需元件

1 × micro:bit 主控板



1 × micro:bit 扩展板



1 × LED 灯模块



1 × 按钮模块



1 × 旋钮模块



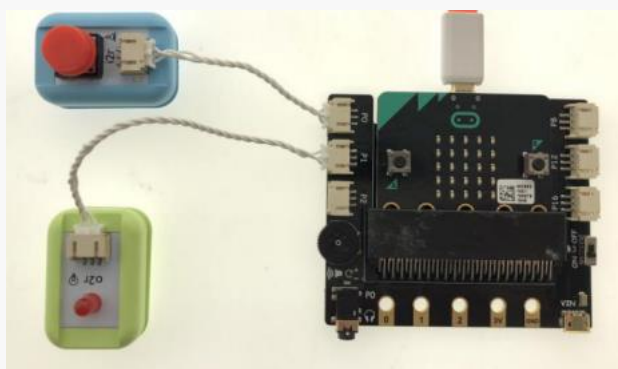
1 × USB A to B 连接线



硬件连接

把按钮模块（旋钮模块）接到扩展板 P0 号接口上；

把 LED 灯模块接到扩展板 P1 号接口上。



编写程序

任务一：按钮台灯

实现功能为：当按下按钮的时候，外接 LED 小灯被点亮，松开按钮以后，外接 LED 小灯熄灭。

STEP1：已将按钮与 micro:bit 的 P0 号引脚相连接，可以通过检测 P0 号引脚的值判断按钮是否被按下，当按钮被按下时，设置 LED 小灯引脚为高电平，LED 灯被点亮，当松开按钮时，设置 LED 小灯引脚为低电平，LED 灯熄灭。按钮模块属于数字输入，单击“进阶”，会出现“Pins”，调用引脚时，通常会运用到这个指令，调用指令的流程为：Pins（进阶指令）----数字读取引脚 P0。



STEP2: 逻辑指令中的“=”可以判断按钮是否被按下，如果按钮值为“1”成立，说明按钮被按下，否则，按钮没有被按下。调用指令的流程为：逻辑----“=”。



STEP3: 如果按钮被按下，外接 LED 小灯被点亮；否则，外接 LED 小灯熄灭。调用指令的流程为：逻辑----如果为（true）一则。



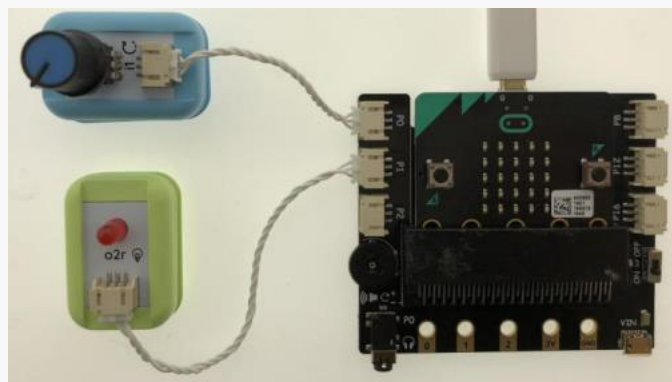
STEP4: 需要用到的指令结合在一起，实现：当按下按钮的时候，外接 LED 小灯被点亮，松开按钮以后，外接 LED 小灯熄灭。



任务二：旋钮控制 LED 灯

实现功能：当旋钮旋至较大值时，micro:bit 控制外接小灯越亮；当旋钮旋至较小值时，micro:bit 控制外接小灯越暗。旋钮的旋转值不同，对应的小灯亮度不同，旋转时，灯的亮度连续变化。

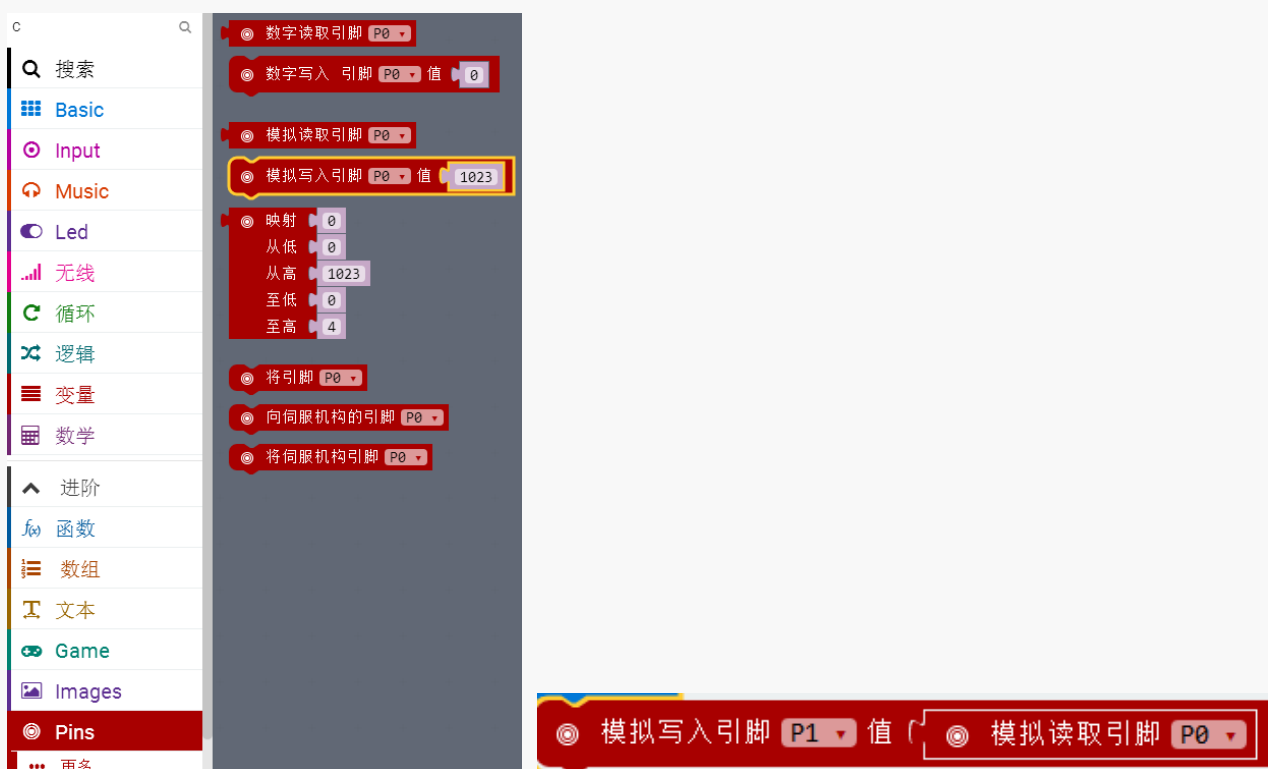
STEP1：硬件连接。把任务一按钮台灯中使用的按钮模块换为旋钮模块。



STEP2: 通过读取旋钮的值，来决定 LED 灯的亮度，旋钮的值为模拟值，取值范围在 0-1023 之间，将其值映射为 LED 灯对应的亮度值。旋钮连接板子的 P0 号引脚，旋钮值调用指令的流程为：Pins（进阶指令）----模拟读取引脚（P0）。



STEP3: LED 灯的亮度由 P0 对应的旋钮值决定，将 P0 的值赋值给 P1，可以控制 LED 灯的改变，调用指令的流程为：Pin（进阶指令）----“模拟写入引脚 P0 值（1023）”，将 P0 改为 P1。之后将获取的旋钮值（P0）放入 P1 值中。



STEP4: 最终程序。



能力提升

利用家里的一些纸盒、纸杯、彩纸等，制作出精致的小灯外形，让它成为你自己的独一无二的作品。

项目四 变速风扇

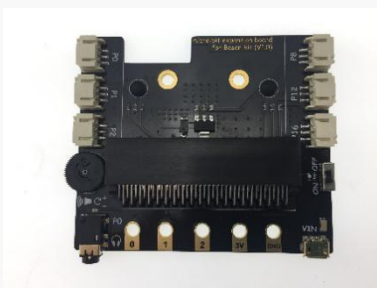
在制作变速风扇之前，先使用之前学习过的按钮，制作一个用按钮控制风扇转和停的装置，了解如何使用风扇，再使用旋钮传感器控制风扇的转速。

所需元件

1 × micro:bit 主控板



1 × micro:bit 扩展板



1 × 风扇模块



1 × 按钮模块



1 × 旋钮模块



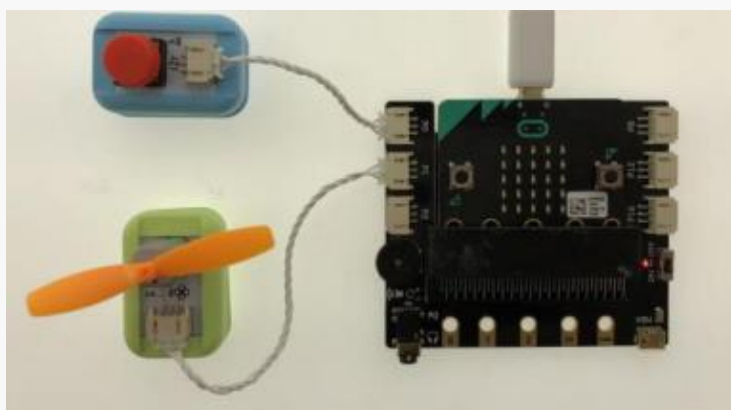
1 × USB A to B 连接线



硬件连接

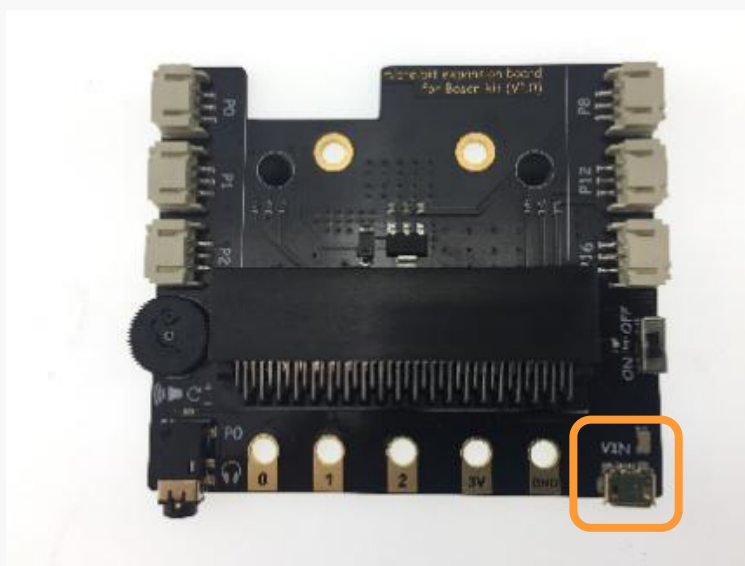
把按钮模块（旋钮模块）接到扩展板 P0 号接口上；

把风扇模块接到扩展板 P1 号接口上。



风扇模块的使用

硬件连接时，我们将 micro:bit 与电脑相连接，实验中，是电脑为 micro:bit 供电，由于风扇或舵机等传感器的耗电量较大，所以，需要外接电源独立供电。外接电源接在扩展板的外接电源接口。

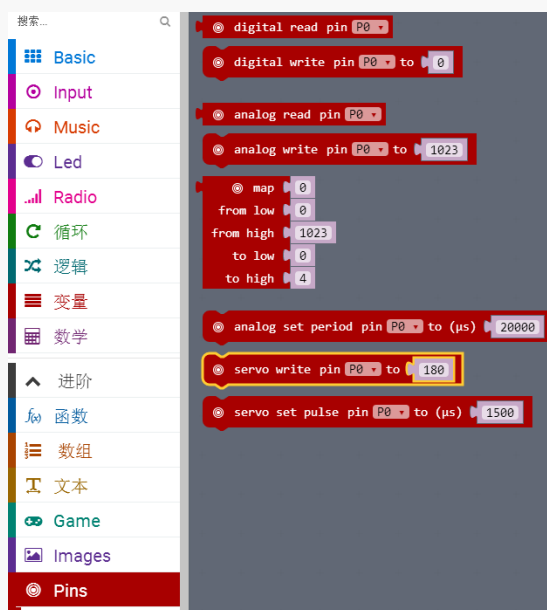


编写程序

任务一：按钮风扇

实现功能：当按下按钮的时候，外接风扇转动，松开按钮以后，风扇停止转动。

STEP1：按钮与 micro:bit 的 P0 号引脚相连接，可以通过检测 P0 号引脚的值判断按钮是否被按下，当按钮被按下时，设置风扇连接引脚为高电平，风扇转动，当松开按钮时，设置风扇连接引脚为低电平，风扇停止转动。按钮模块属于数字输入，单击“进阶”，会出现“Pins”，调用引脚时，通常会运用到这个指令，调用指令的流程为：Pins（进阶指令）---数字读取引脚 P0。



STEP2：逻辑指令中的“=”可以判断按钮是否被按下，如果按钮值为“1”成立，说明按钮被按下，否则，按钮没有被按下。调用指令的流程为：逻辑----“=”。



STEP3: 如果按钮被按下，风扇转动；否则，风扇停止转动。调用指令的流程为：逻辑----如果为（true）----则----否则。



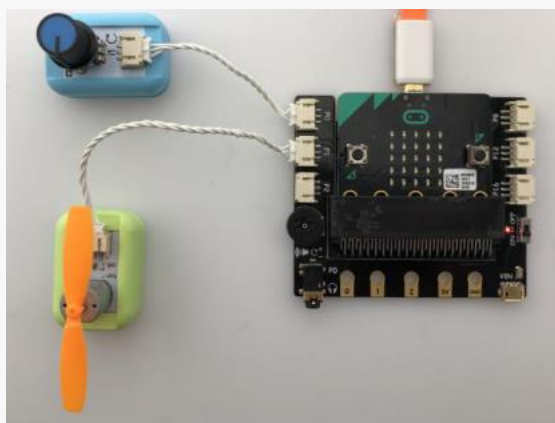
STEP4: 需要用到的指令结合在一起，实现：当按下按钮的时候，外接风扇转动，松开按钮以后，风扇停止转动。



任务二：变速风扇

实现功能：当旋钮旋至较大值时，micro:bit 控制风扇转速增大；当旋钮旋至较小值时，micro:bit 控制风扇转速变小。旋钮的旋转值不同，对应的风扇转速不同。

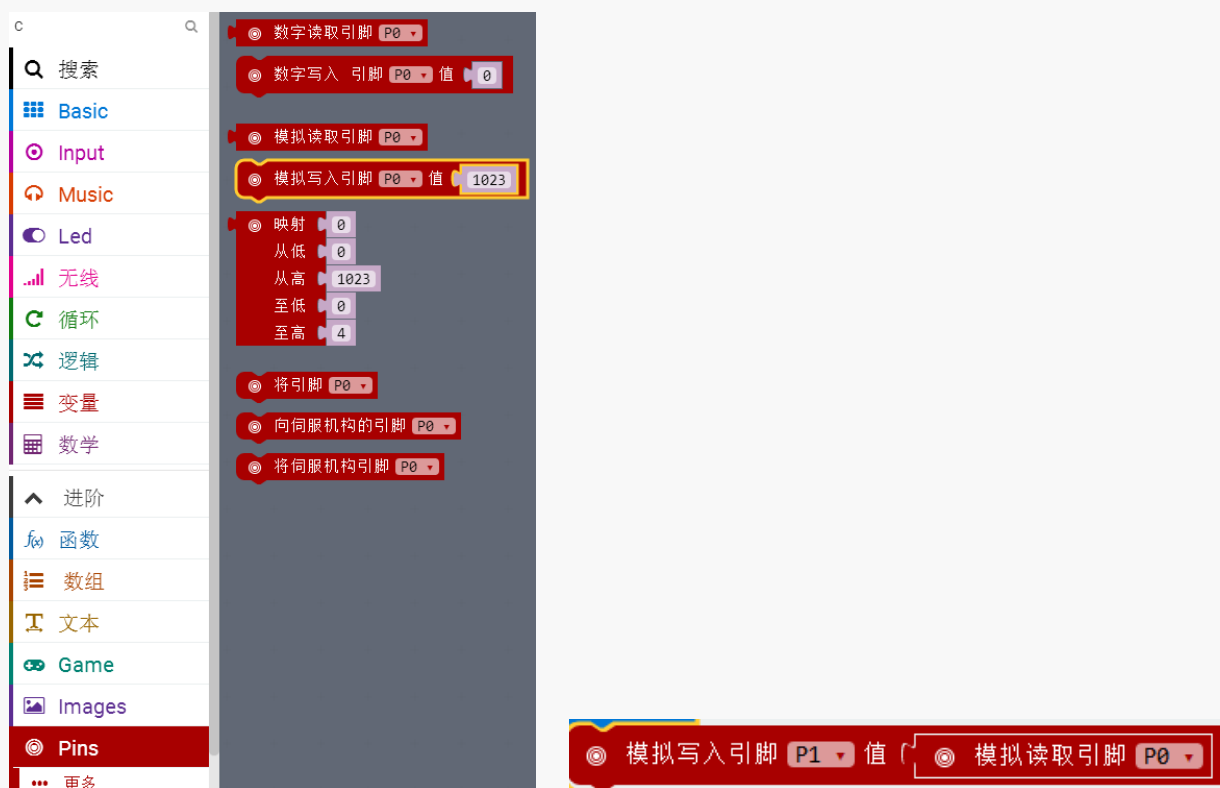
STEP1：硬件连接。将任务一按钮风扇用到的按钮模块换成旋钮模块。



STEP2：硬件连接中运用到了风扇，通过读取旋钮的值，来决定风扇的转速，旋钮的值为模拟值，取值范围在 0-1023 之间，将其值映射为风扇对应的转速。旋钮连接板子的 P0 号引脚，旋钮值调用指令的流程为：Pins（进阶指令）---模拟读取引脚（P0）。



STEP3: 风扇的转速由 P0 对应的旋钮值决定，将 P0 的值赋值给 P1，可以控制风扇转速的改变，调用指令的流程为：Pins（进阶指令）---- “模拟写入引脚 P0 值（1023）”，将 P0 改为 P1。之后将获取的旋钮值（P0）放入 P1 值中。



STEP4: 最终程序。



第四话：如虎添翼

如虎添翼

在学习了按钮模块、旋钮模块控制输出端之后，你是不是觉得好像还少了点什么呢？我们常说的传感器还有很多，在这一章节的内容学习中，我们将认识更多的传感器使用，也将制作出更加智能化、好玩和实用的作品。

项目一 电子蜡烛

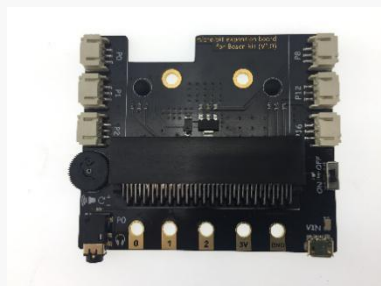
在这一项目中，我们要制作一个模拟蜡烛的装置，减少碳排放，为环保事业做贡献，我们以后过生日的时候就可以自己制作一个电子蜡烛啦！还能真的吹灭电子蜡烛哦！

所需元件

1 × micro:bit 主控板



1 × micro:bit 扩展板



1 × LED 灯模块



1 × 声音传感器模块



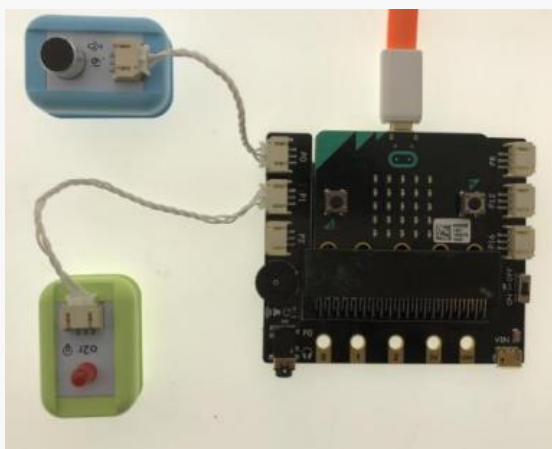
1 × USB A to B 连接线



硬件连接

把声音传感器模块接到扩展板 P0 号接口上；

把 LED 灯模块接到扩展板 P1 号接口上。



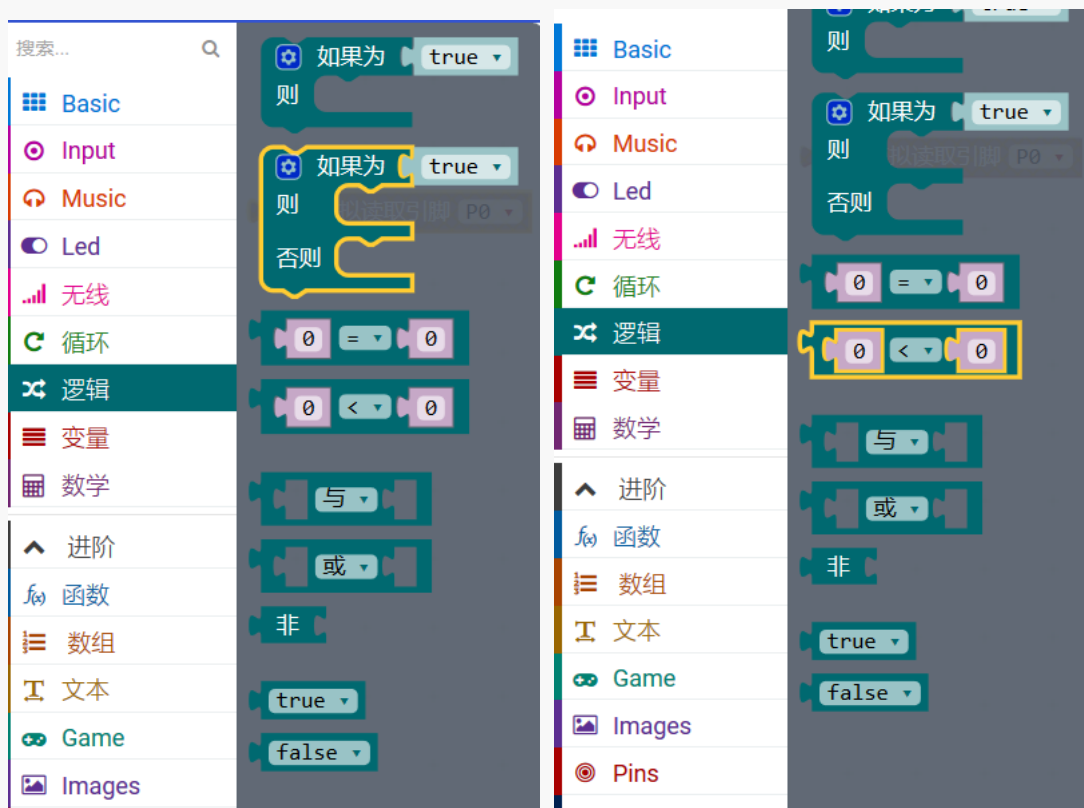
编写程序

要想制作电子蜡烛，首先要认识声音传感器。我们可以将声音传感器与 micro:bit 板连接，通过显示板载数字来观察我们的声音强弱程度。

任务一：声音强度探测

实现功能：当声音强度模拟值小于 50 的时候，板载小灯显示数字“1”，表示声音较小，强度为 1 级；否则，声音强度模拟值大于 50 的时候，板载小灯显示数字“2”，表示声音较大，强度为 2 级。

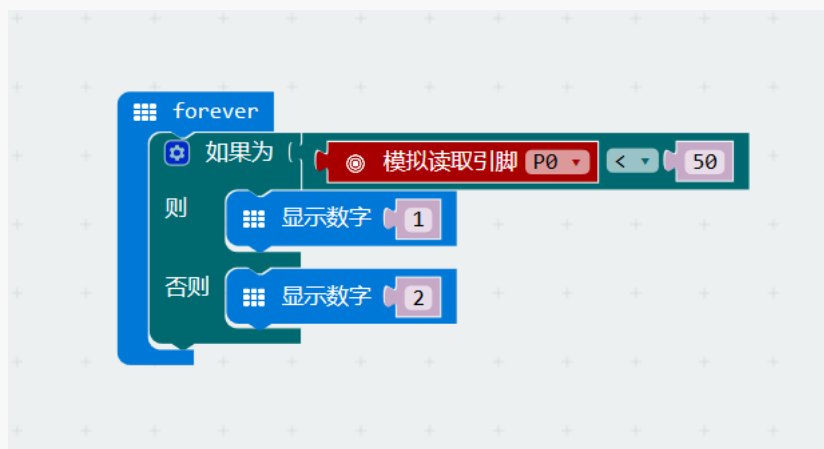
STEP1: 硬件方面我们暂时用不到 LED 灯模块。将逻辑中的条件语句代码模块和逻辑判断数值大小的代码模块指令结合拖动入“forever”中，条件语句代码模块的功能在于设定触发显示板载数字的条件，如果满足右边逻辑判断语句。则执行在“则”右侧的代码语句，反之，如果不满足逻辑判断语句，则执行的是在“否则”右侧的代码。



STEP2: 模拟读取引脚读取 P0 引脚连接的声音传感器的模拟信号数值，同之前讲过的一样，模拟数值范围在 0—1023 之间。



STEP3: 设定“声音强度探测”的实现程序。逻辑指令中的“<”可以判断声音强度的模拟值是否达到设定的标准，如下图，表示模拟读取引脚 P0 的数值“<50”成立的时候，声音强度为 1 级的时候，板载小灯显示数字“1”，否则，模拟读取引脚 P0 的数值“>50”时，声音强度为 2 级，板载小灯显示数字“2”。



对着声音传感器说话或者吹气，看你的板载 LED 点阵会不会显示相应的数字。

如果你已经很用力说话或吹气了却还是不能显示相应的数字，该怎么调节临界值 50 这个强度呢？改为更大还是更小？

任务二：电子蜡烛

实现功能：当在 Makecode 在线网站编写程序后，烧录至 Micro: bit 小板子，通过声音传感器控制 LED 小灯的亮灭，就像过生日的时候吹蜡烛一样，对着声音传感器吹气，当声音达到一定强度的时候，比如之前设定的 2 级强度时，小灯就熄灭。

STEP1: 编程过程中，通过设置 Micro: bit 的 p2 引脚值（连接声音传感器的引脚）来控制小灯的亮暗。当声音达到设定的强度的时候，小灯熄灭；反之，当声音未达到设定强度时候，小灯保持亮起的状态。将如图所示的判断语句代码模块指令拖动入“forever”中。



STEP2: 设定“电子蜡烛”的实现程序。当声音传感器模拟值小于设定值 50 的时候，小灯保持亮起的状态，反之，轻轻一吹，声音强度大于 50 时，小灯将保持 3 秒的熄灭状态。



项目二 自动门

生活中我们常常那个可以看见这样的装置：当我们靠近便利店大门的时候，门就自动打开了，有时还会伴随着“欢迎光临”的声音。

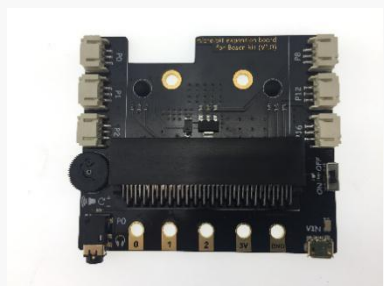
本节课制作的项目就是要来模拟这样的自动门装置，我们可以使用运动传感器感应有人过来了，用舵机模拟开门的动作，接下来，我们开始制作吧！

所需元件

1 × micro:bit 主控板



1 × micro:bit 扩展板



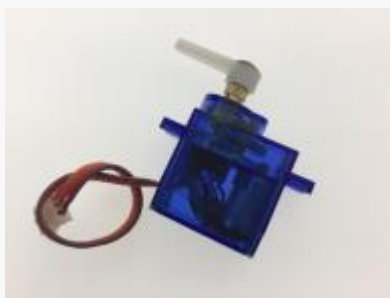
1 × 按钮模块



1 × 运动传感器模块



1 × 舵机



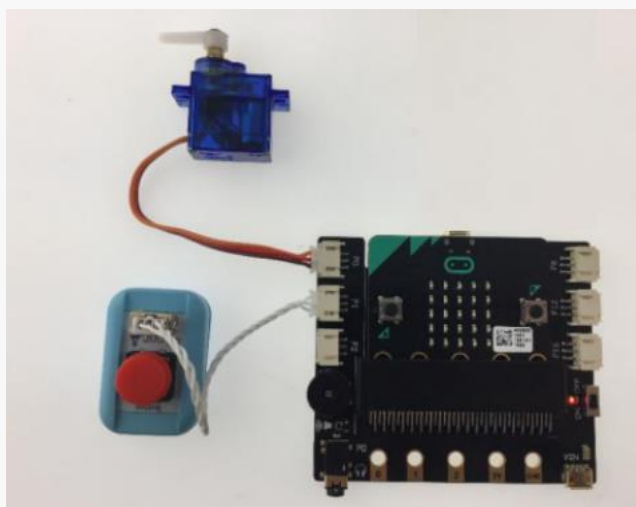
1 × USB A to B 连接线



硬件连接

把舵机接到扩展板 P0 号接口上；

把按钮模块（运动传感器）接到扩展板 P1 号接口上。

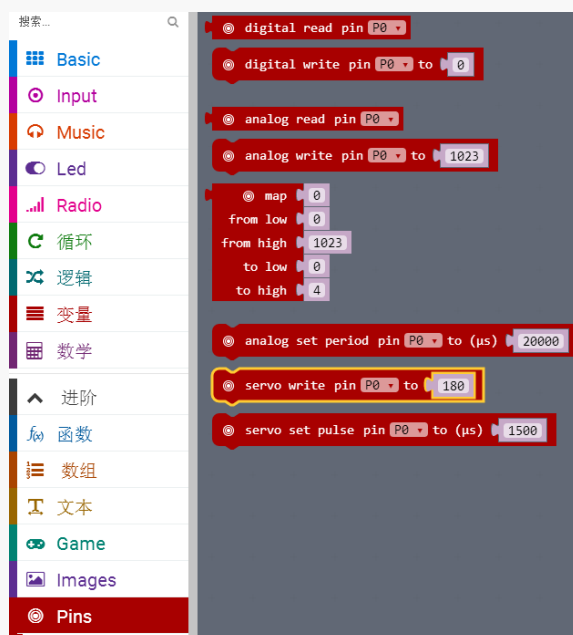


编写程序

任务一：按钮控制舵机

实现功能：当按下按钮的时候，舵机转动 100° 并且 micro:bit 显示 “O”，按钮没有被按下时，舵机旋转 0° 并且 micro:bit 显示 “X”。

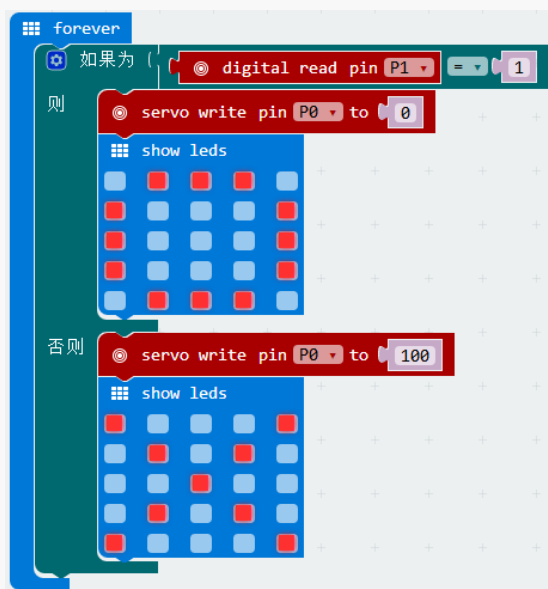
STEP1：已将舵机与 micro:bit 的 P0 号引脚相连接，通过调用 P0 号引脚可以读取舵机的值，调用指令的流程为：pins（进阶指令）----servo write pin(p0) to(100)。（舵机连接 P0 引脚，故不用修改引脚值）



STEP2: 如果按钮按下，实现舵机转动 100° 并且 micro:bit 显示 “O”；否则舵机旋转 0° 并且 micro:bit 显示 “X”。需要用到的指令为：如果为 (true) --则--否则。



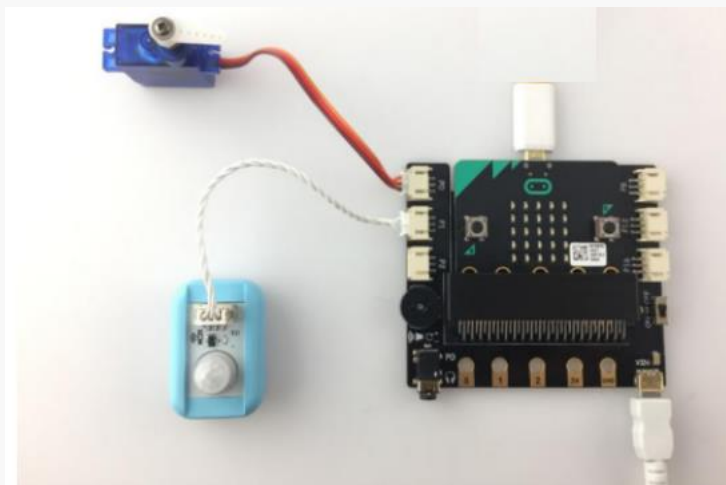
STEP3: 最终程序。注意：由于舵机耗电量大，需要外接电源。



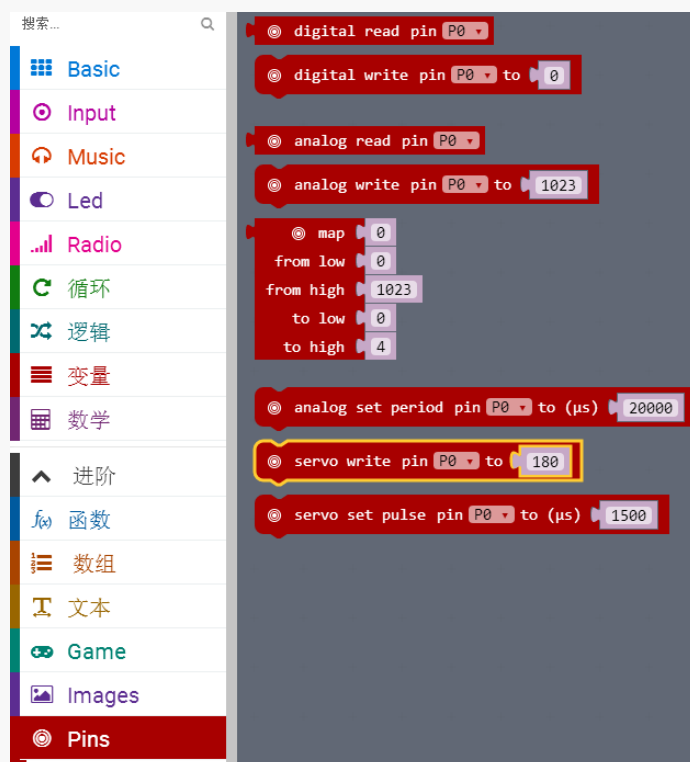
任务二：运动传感器控制舵机

实现功能：当运动传感器检测到有人经过时，舵机转动 100° 并且 Micro:bit 显示“O”，否则，舵机旋转 0° 并且 micro:bit 显示“X”。

STEP1：硬件连接。将任务一按钮控制舵机中的硬件按钮模块换成运动传感器模块。



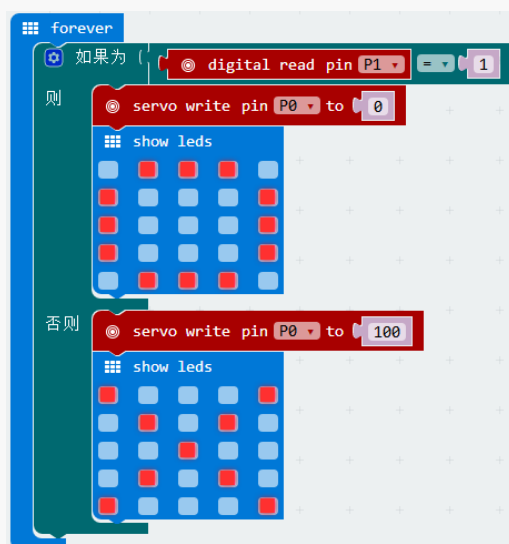
STEP2：运动传感器与 micro:bit 的 P1 号引脚相连接，通过调用 P1 号引脚可以读取运动传感器的值，调用指令的流程为：pins（进阶指令）----digital read pin p0。（拖动后，将 P0 改为 P1）



STEP3: 当运动传感器检测到有人经过时，实现舵机转动 100° 并且 micro:bit 显示 “O”；否则，舵机旋转 0° 并且 micro:bit 显示 “X”。需要用到的指令为：如果为 (true) --则--否则。



STEP4: 最终程序。



项目三 音乐盒

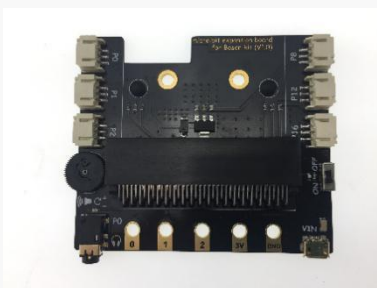
在礼品商店里，我们总能看到形状各异，非常吸引眼球的音乐盒，它不仅长得好看，还非常有趣，会为我们播放动听的音乐。这节课要制作的作品就是会播放音乐的音乐盒，话不多说，赶紧行动起来吧！

所需元件

1 × micro:bit 主控板



1 × micro:bit 扩展板



1 × 普通耳机



1 × 运动传感器模块



1 × USB A to B 连接线



硬件连接

用数据线将 micro:bit 主板与电脑连接起来，将耳机插入耳机插孔；

将运动传感器模块接在扩展板的 P1 号接口。

注意：播放音频的耳机插孔数据通信占用了 P0 端口，因此播放声音的时候，P0 端口不可用。



编写程序

任务一：电子音乐盒

实现功能：让 micro:bit 主板播放音乐。

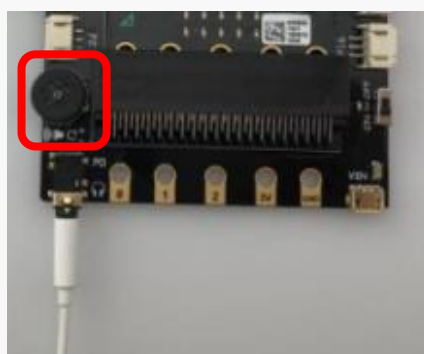
STEP1：将 music 模块集合中的播放旋律模块，拖动到 forever 中。播放旋律模块中已经有内置的很多旋律可供选择，我们在这里选择的是第一段旋律 “dadadum” 重复一次。



STEP2: 将选择的旋律“dadadum”一段完整的占 4 个节拍，所以在播放旋律模块的下方需要设定旋律播放的时长为 4 个节拍。最终程序。



STEP3: 在扩展板耳机插孔的上方紧挨着的“小转盘”，试着旋转它，可以调节 micro:bit 板播放声音的大小。



任务二：智能音乐盒

实现功能：当有人靠近时，让 micro:bit 主板播放自己编的一段音乐；当没有人靠近时，不播放音乐。

STEP1: 编写一段歌曲《小星星》的音乐。

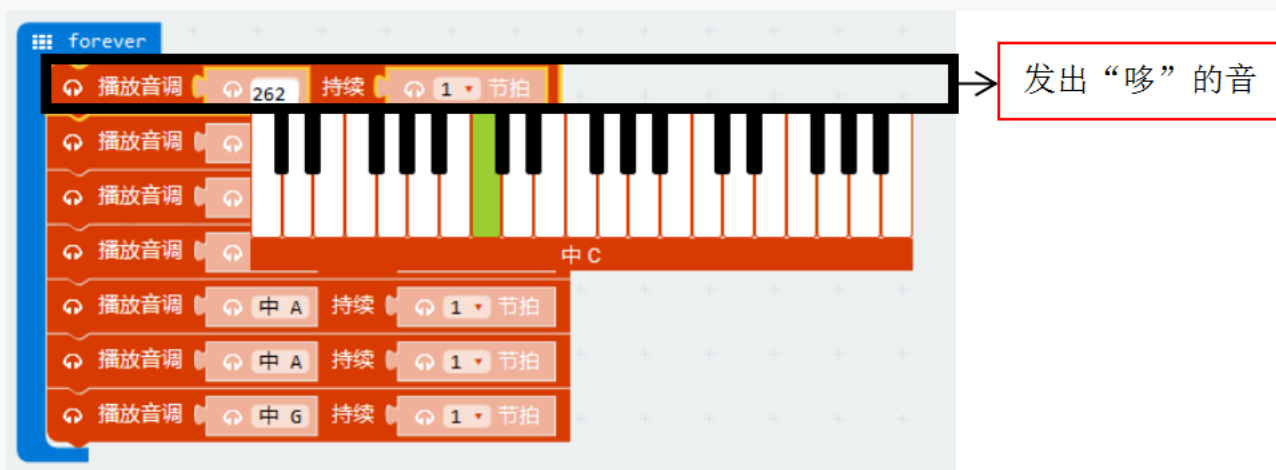
先尝试编写能演奏小星星第一句的代码，先看看这首歌的简谱是怎样的，音调不同，音符不同，需要的发声频率也不同。



以中音 C 调为例，发声频率与其对应的音符如下表所示：

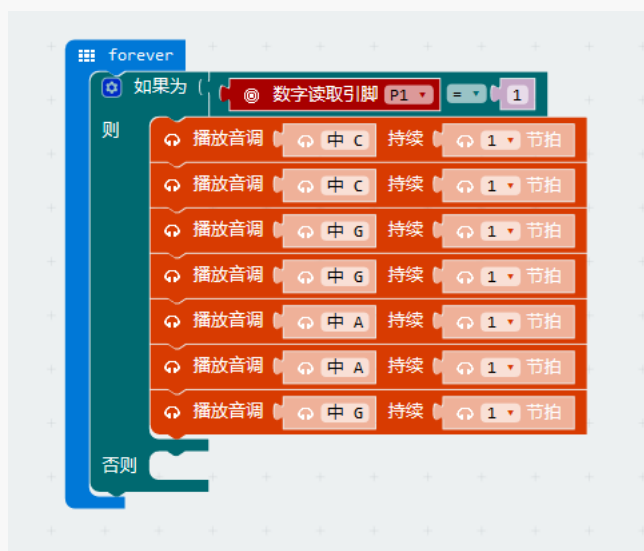
音符	1	2	3	4	5	6	7
频率	262	294	330	350	393	441	495

打开 music 代码集合中的代码模块，现在用到的是播放音调模块。在这里可以设置发出的音调和所占节拍。



编写好小星星的曲子之后我们的 micro:bit 已经学会唱歌啦，但是如何让它变得智能呢？这时候，我们就需要用到运动传感器了。其实很简单，只需要添加一个我们之前已经学过的条件判断语句就可以让智能音乐盒得以实现啦！一起来看看代码是怎样的吧？

STEP2：最终程序。



项目四 炫彩灯带

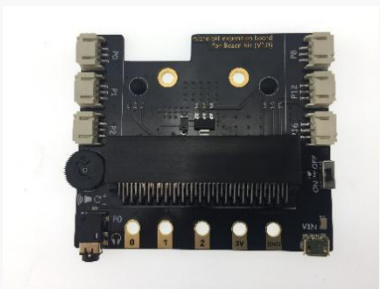
上节课的项目音乐盒做完之后是不是感觉缺少点什么呢？我们生活中见到的音乐盒还会伴随着非常炫酷的灯光登场，那么这节课我们就要学习制作炫酷的彩虹灯啦！是不是迫不及待了呢？我们开始进入奇幻的灯光世界吧！

所需元件

1 × micro:bit 主控板



1 × micro:bit 扩展板



1 × 彩虹灯带



1 × 声音传感器模块

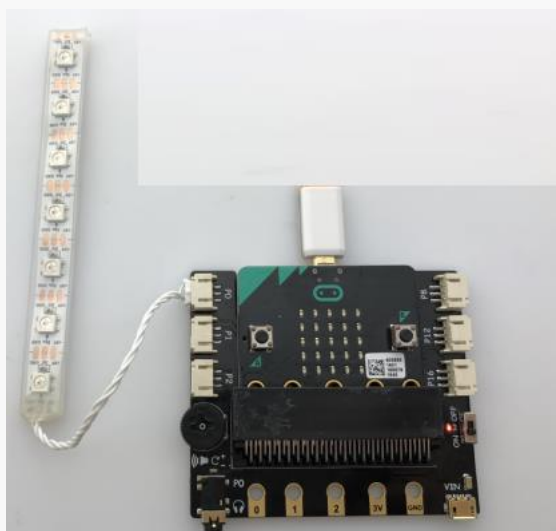


1 × USB A to B 连接线



硬件连接

将彩虹灯带接在扩展板的 P0 号接口。

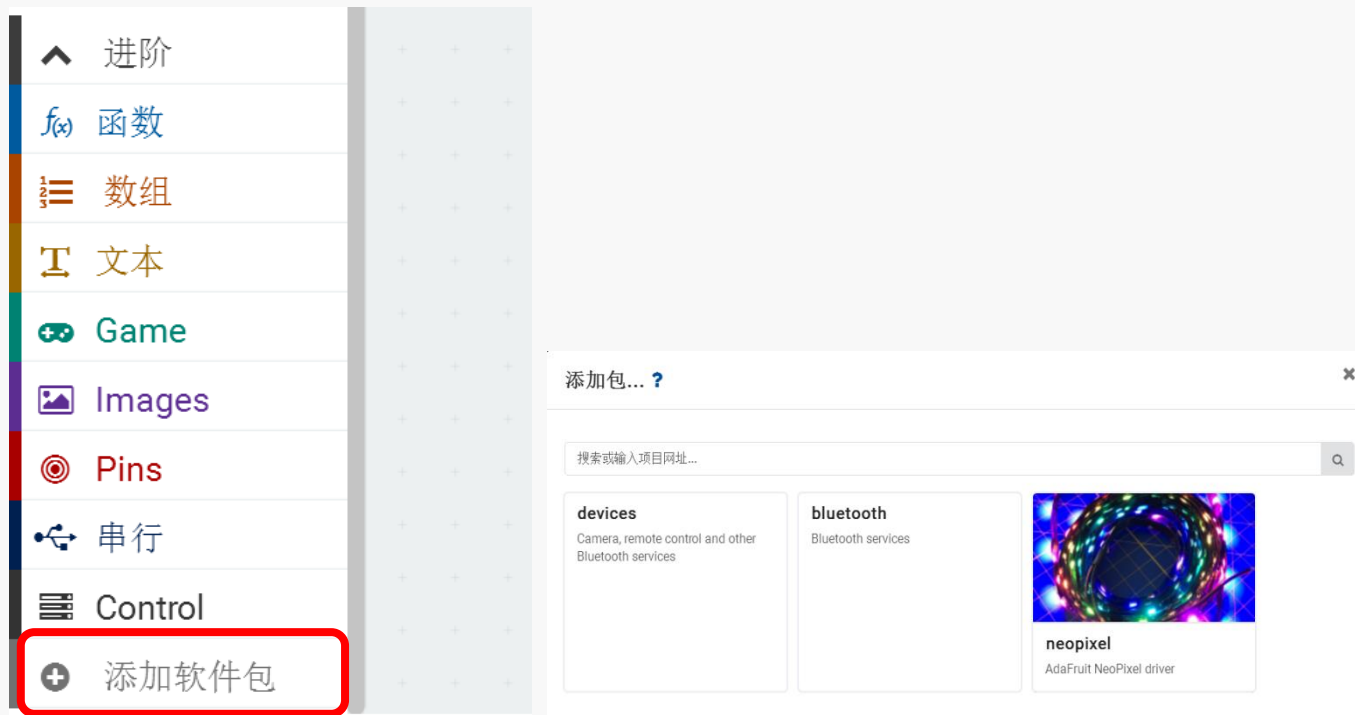


编写程序

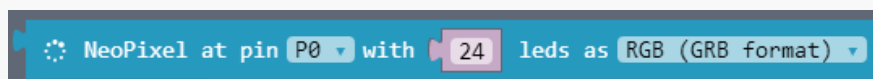
任务一：点亮彩虹灯带

实现功能：通过编写程序，将程序烧录至板子后，可以控制将彩虹灯带点亮并呈现 7 彩的效果。

STEP1:平时看到的指令区的模块中,没有加载使用灯带的指令库,使用灯带需要调用 Neopixel 指令,Neopixel 指令的调用需要滚动至指令区的最低部,单击“添加软件包”,出现“添加包”的菜单栏,调用指令的流程为:进阶----添加软件包----neopixel。单击 neopixel 会增加 neopixel 模块。



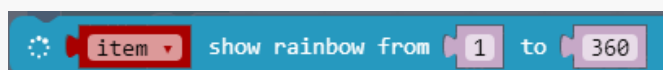
STEP2: 平时看到的指令在使用彩虹灯带的过程中，首先选定灯带连接 micro:bit 的引脚、灯带 LED 灯的数目及颜色模式，需要调用的指令如下：



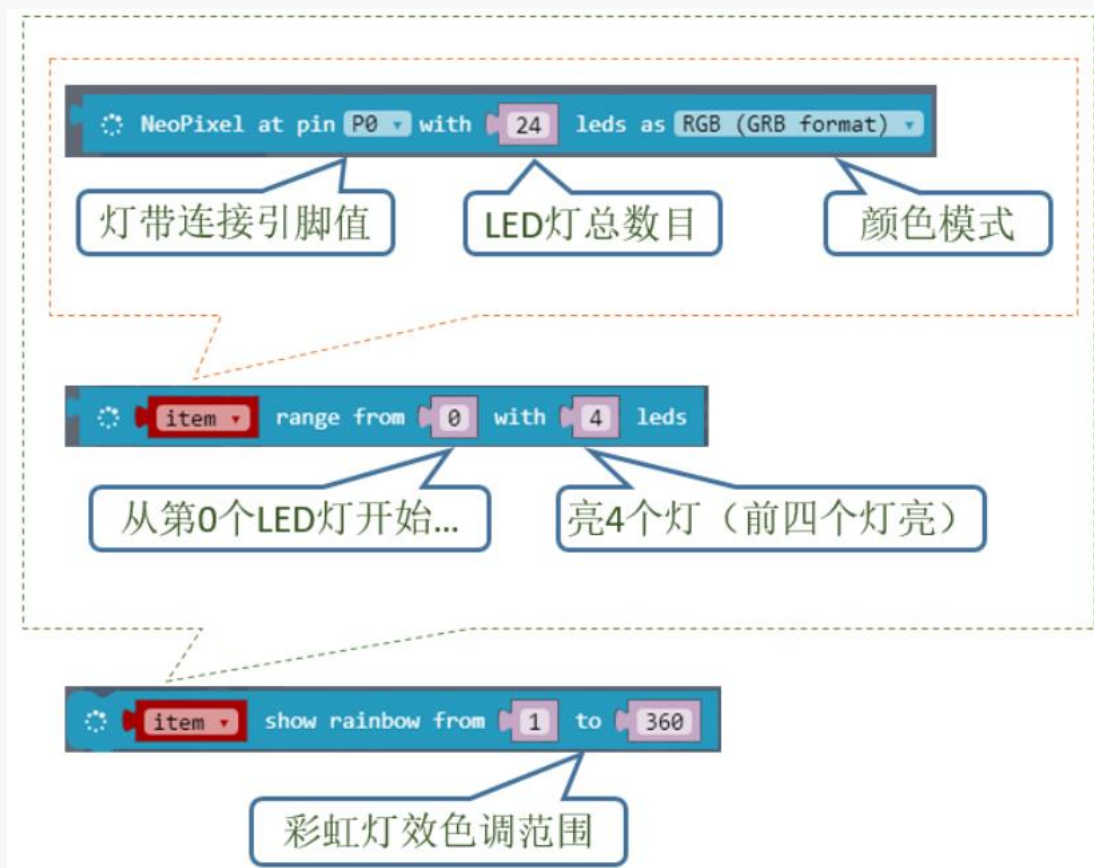
之后要确定 LED 灯使用的范围，需要调用的指令是：



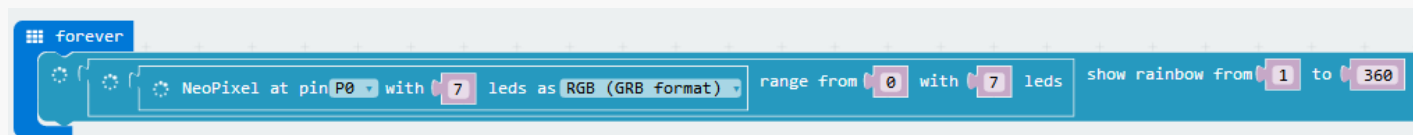
最后要确定彩虹灯效色调范围，需要调用的指令是：



使用灯带的相关指令及说明如下：（在使用过程中，对参数会进行修改）



STEP3: 灯带连接引脚为 P0，包含 7 个 LED 小灯，设置彩虹灯效色彩范围 1-360（红色--绿色--蓝色），彩灯呈现七彩效果。调整参数后，最终程序。



任务二：依次点亮彩虹灯带

实现功能: 通过编写程序，将程序烧录至板子后，可以控制彩虹灯带的 LED 小灯每隔一秒点亮一盏 LED 小灯，呈现炫彩灯效。

STEP1: 控制彩虹灯带的 LED 小灯每隔一秒点亮一盏 LED 小灯，需要运用可以存放变量的指令，每次来控制小灯被点亮的具体数目，需要调用指令的流程为：变量----设置变量，任意设定变量名（如：点亮小灯数），再单击确定。



New variable name:

确定 ✓

取消 ✕

New variable name:

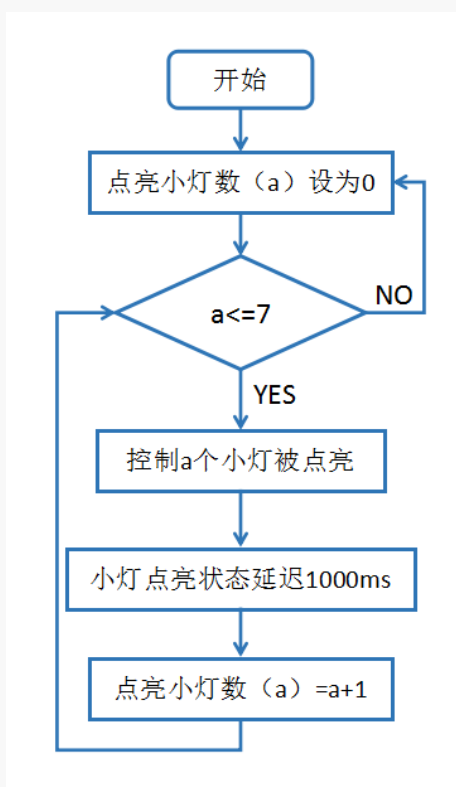
确定 ✓

取消 ✕

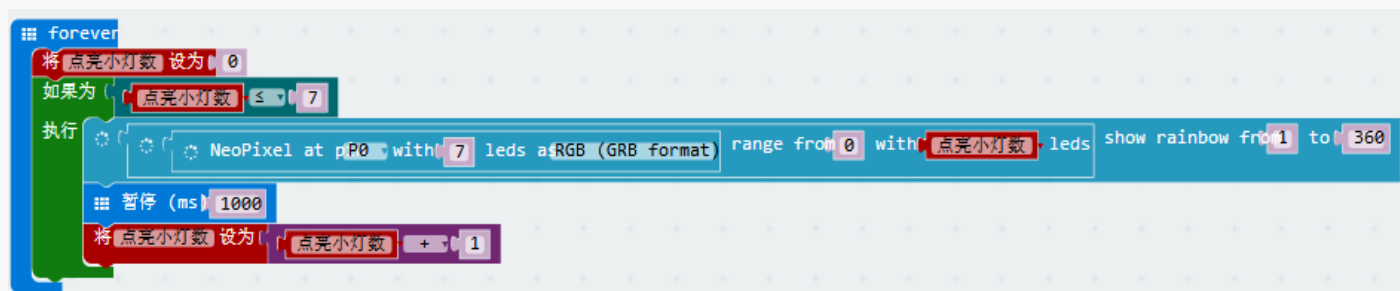
STEP2: “点亮小灯数”（也可以简称为“a”）最初设定变量的初值为 0（最初没有小灯被点亮），然后控制变量在原有的基础上增加 1，需要运用到循环，当“点亮小灯数”不到 7 个的时候，继续点亮小灯，需要调用的指令如下。



STEP3: 当“点亮小灯数”(a) 不到 7 时, 可以点亮 a 个小灯, 延迟 1000ms, 之后让小灯数增加 1, 流程图如下。



STEP4: 将点亮小灯数 (α) 最初设置为 0, 点亮小灯数 (α) ≤ 7 时, 控制点亮 α 个小灯, 并延迟 1000ms, 接着将点亮小灯数增加 1, 如此循环, 直到点亮小灯数大于 7, 所有小灯熄灭, 控制彩虹灯带的 LED 小灯每隔一秒点亮一盏 LED 小灯。



任务三：声音控制彩虹灯带

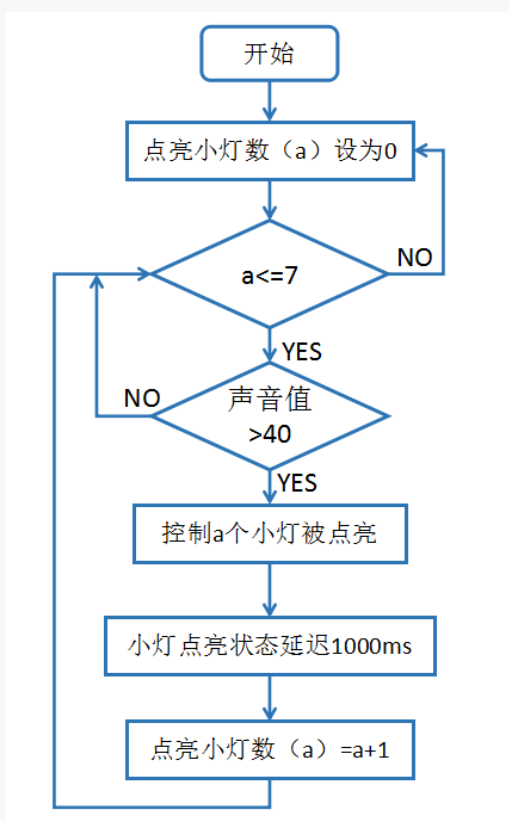
实现功能：最初灯带为熄灭状态，当发出声音时，点亮一盏 LED 灯，当没有声音时，灯带保持点亮部分小灯状态。

STEP1: 这个任务需要运用到声音传感器，当发出声音时，声音传感器可以检测到，控制灯带点亮 LED 小灯。之前章节学过声音传感器的用法，我们再来巩固一下调用方法，调用指令的流程为：Pins（进阶指令）----模拟读取引脚（p0），将 P0 修改为 P1。

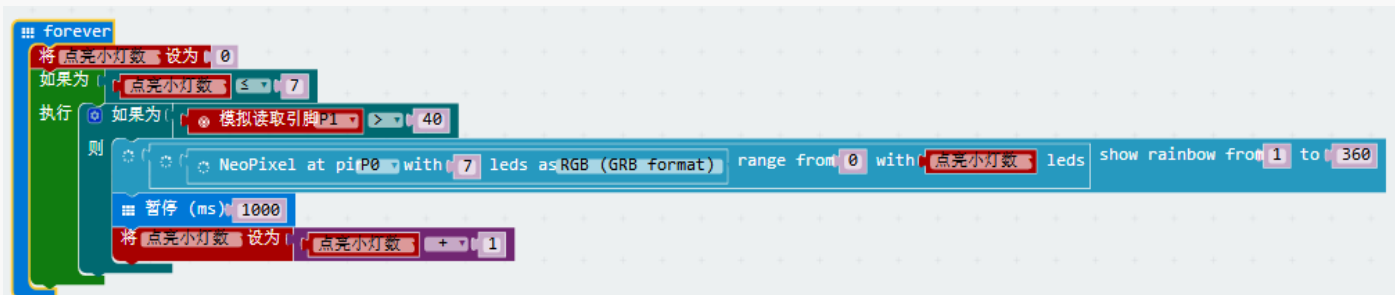


STEP2: 当声音传感器检测到声音 > 40 且点亮灯的数量不超过 7 盏时，控制灯带点亮一盏 LED 灯，之后持续不断检测声音是否超过 40。

流程图：



STEP3: 最终程序。



第五话：大展拳脚

大展拳脚

经过前面这么多个项目的学习，你是不是对 micro:bit 已经了如指掌了呢？在这一章节的内容学习中，你会更高层次地了解 microbit，也会制作出非常炫酷好玩的作品！

项目一 电子稳定器

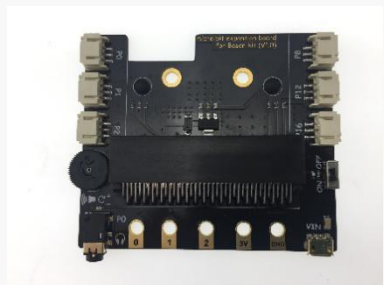
你知道什么是重力传感器吗？其实重力传感器的运用是非常常见的，它并不是那么遥不可及，比如说，我们常使用的手机或平板电脑，你本来把手机拿在手里时是竖着的，你将它转 90 度，横过来，它的页面就跟随你的重心自动反应过来，也就是说页面也转了 90 度，这是一项极具人性化的设计。这节课的项目我们就是要来模拟这样的功能哦！

所需元件

1 × micro:bit 主控板



1 × micro:bit 扩展板



1 × 舵机

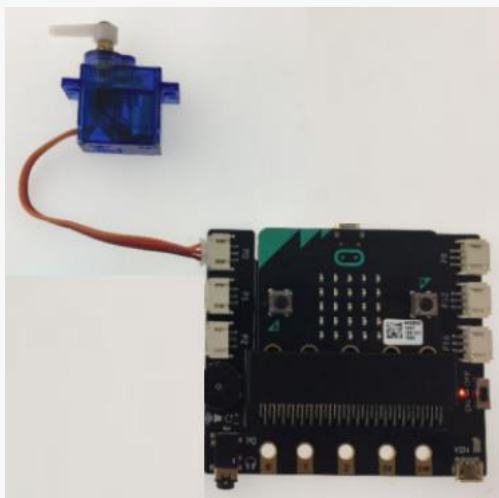


1 × USB A to B 连接线



硬件连接

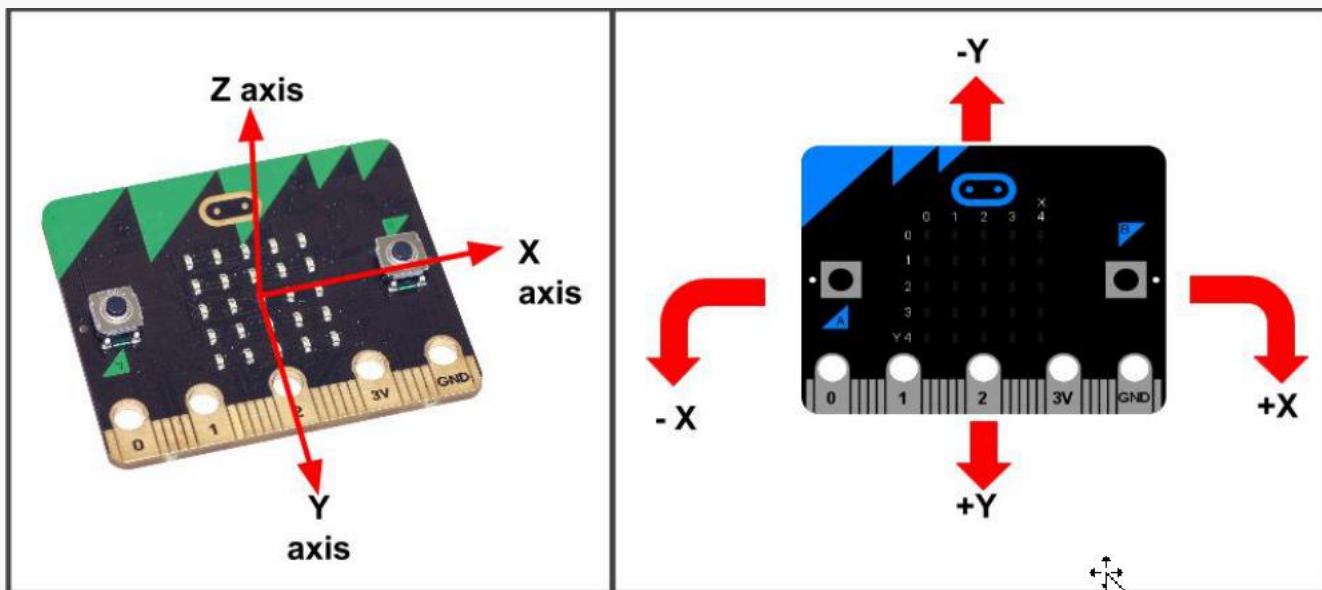
将舵机与 micro:bit 的 P0 号接口相连接。



编写程序

实现功能：通过编写程序，将程序烧录至板子后，可以让舵机指针保持指向竖直上方。

Micro:bit 能实时通过重力传感器自动检测姿态，从而控制舵机一直指向正上方。Micro:bit 可以检测 X,Y,Z 三个方向的重力加速度，其中 X 沿着左右方向，Y 沿着前后方向，Z 垂直于板，沿着上下方向。



X,Y,Z 轴的数值可以在功能模块 “输入” 下找到。

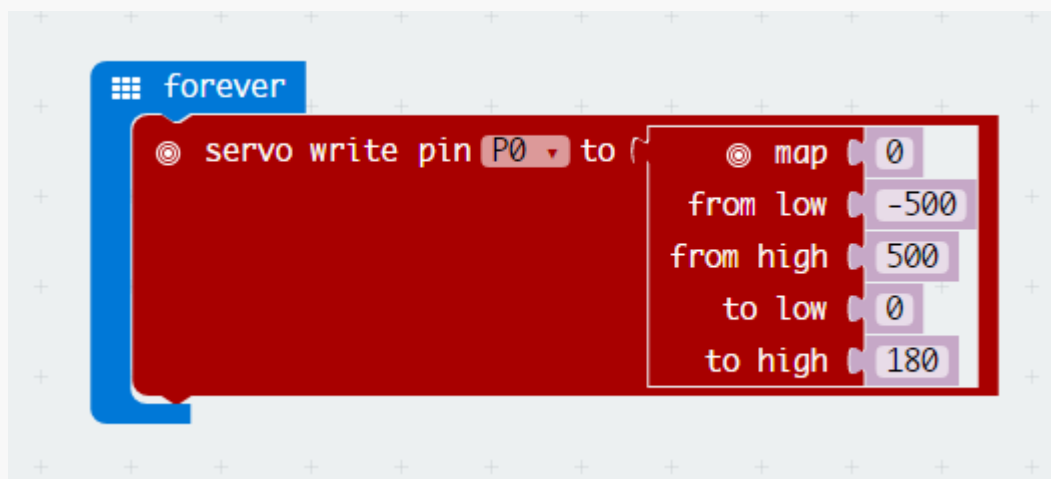


当主板平放在桌面上时，micro:bit 只受到上下方向上的重力，所以 X,Y 都接近于 0，而 Z 轴则约 1000(mm/S2),大约等同于重力加速度)。

而当板子产生倾斜时，X 和 Y 的数值则会发生变化。向左侧倾斜则 X 为负值，反之为正值。

所以可以通过获取 micro:bit 的倾斜角度，控制舵机旋转一个反向的角度，让舵机的指针一直朝上。

STEP1：当板子向左倾斜时，重力传感器的 X 方向会产生一个负向的输出值，我们将 micro:bit 向 0 度旋转，来抵消这个倾斜夹角，反之亦然。由于重力传感器的数值范围在-2048~2048 之间，而-500 到 500 是一个相对有效的检测区域，我们可以通过 map 模块，将-500~500 分别对应到舵机的 0~180 度角度。



STEP2: 用双面胶将舵机和扩展板固定起来, 使两者保持相对静止。注意舵机头的朝向需要与 USB 口的方向一致, 这样才能保证齿轮的转轴垂直于 X 轴, 让舵机齿轮的旋转与 micro: bit 的倾斜方向互补。

能力提升

尝试使用 micro:bit 自带的电子罗盘功能制作一个指南针。

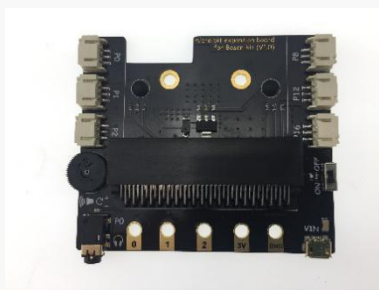
项目二 DJ 演奏台

所需元件

1 × micro:bit 主控板



1 × micro:bit 扩展板



1 × 彩虹灯带



1 × 旋钮模块



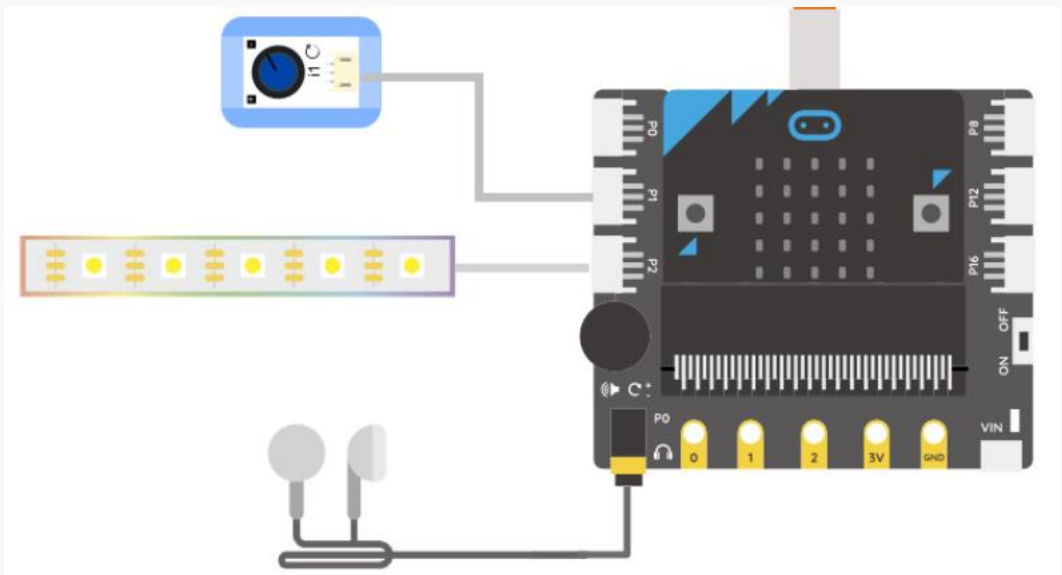
1 × USB A to B 连接线



硬件连接

将旋钮模块接在扩展板的 P1 号接口；

将彩虹灯带接在扩展板的 P2 号接口。



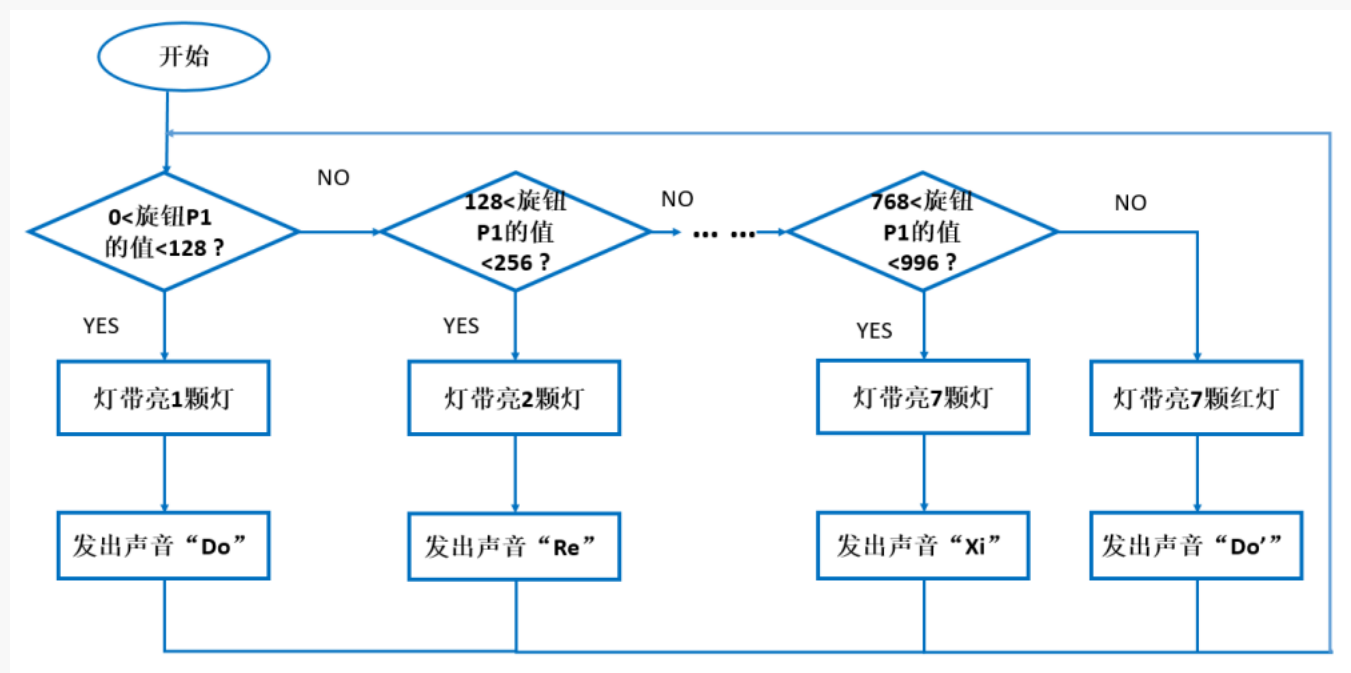
编写程序

实现功能：当用手调整旋钮的转动角度时，实现音乐演奏，同时配合灯带的效果实现。

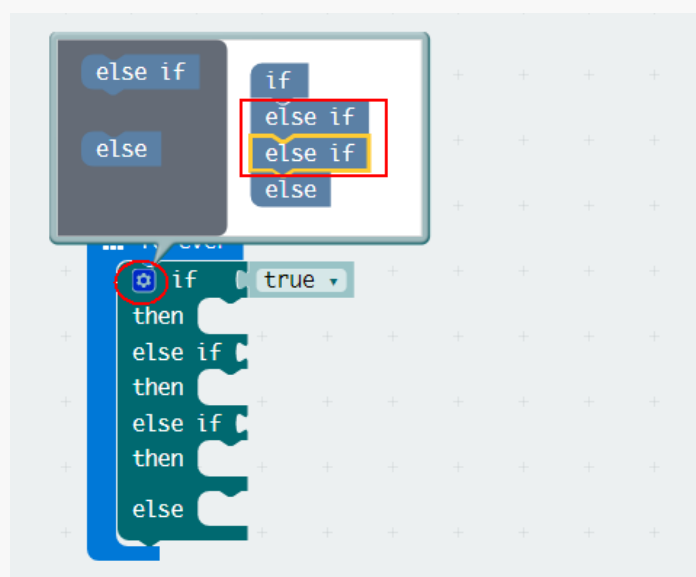
当用手拨动旋钮时，对应的 P1 号引脚读出的模拟值在 0~1023 范围变化，分隔成 8 个小范围，对应音乐中的一个八度“Do~Do'”，对应灯带上个数的变化。

旋钮（P1）的输入值	声音	灯带
0~128	Do	1 颗灯亮
128~256	Re	2 颗灯亮
256~384	Mi	3 颗灯亮
384~512	Fa	4 颗灯亮
512~640	So	5 颗灯亮
640~768	La	6 颗灯亮
768~996	Xi	7 颗灯亮
996~1023	Do'	7 颗灯亮，都为红色

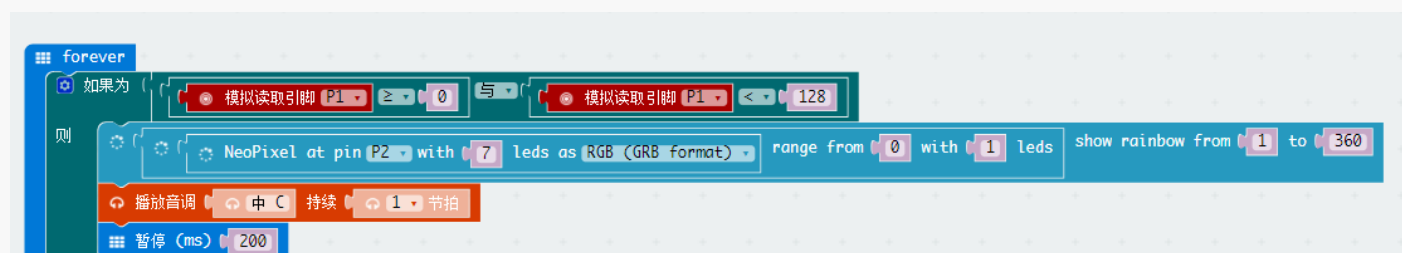
STEP1: DJ 演奏台的实现逻辑图。



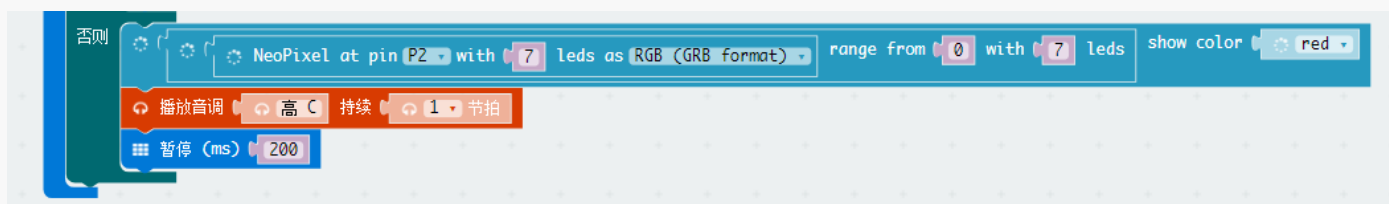
STEP2: 逻辑模块“if”的编辑，逻辑模块可以进行以下编辑操作。



STEP3: 判断旋钮输入 (P1) 的值的范围，对应灯带与音效。以下是第 1 个区域的控制程序。



STEP4: 由于只有 7 颗灯，在第 8 个区域中，可以通过不同的灯带表现形式来表达 “Do’ ”。



STEP5: 最终程序。



能力提升

把 micro:bit 主板的 5x5LED 点阵也利用起来，当演奏出不同的音节时，对应显示该音节。尝试一下制作一下吧！

项目三 可移动门铃

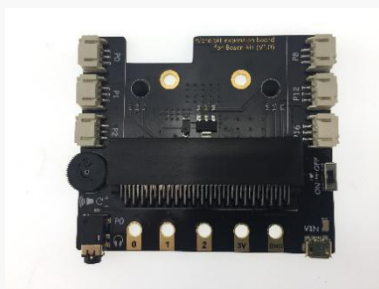
门铃，是我们生活中很常见的东西，几乎家家户户都有，它的作用在于客人来到家里方便提醒主人开门迎接的，但是有时候我们在做其他事情的时候常常会听不到门铃声，有什么办法可以解决这个问题呢？在这个项目中，我们将会学习如何制作可以远程移动的门铃。

所需元件

1 × micro:bit 主控板



1 × micro:bit 扩展板



1 × LED 灯模块



1 × 运动传感器模块



1 × 按钮模块

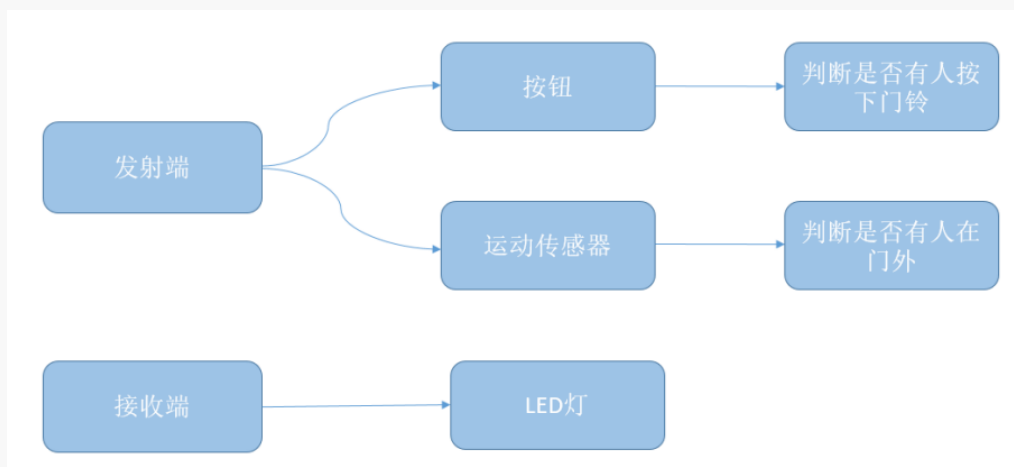


1 × USB A to B 连接线

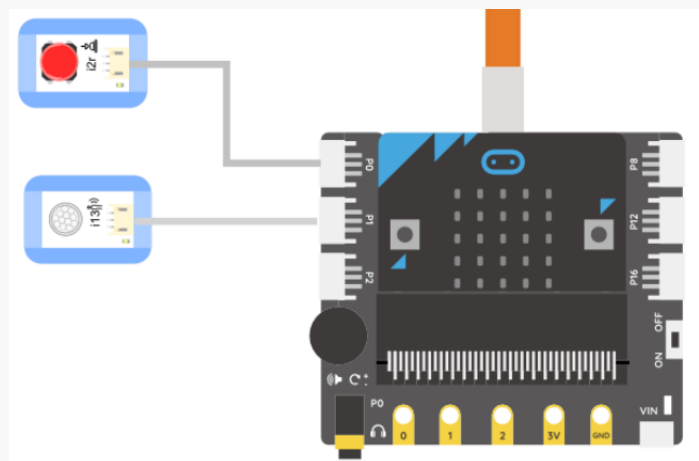


硬件连接

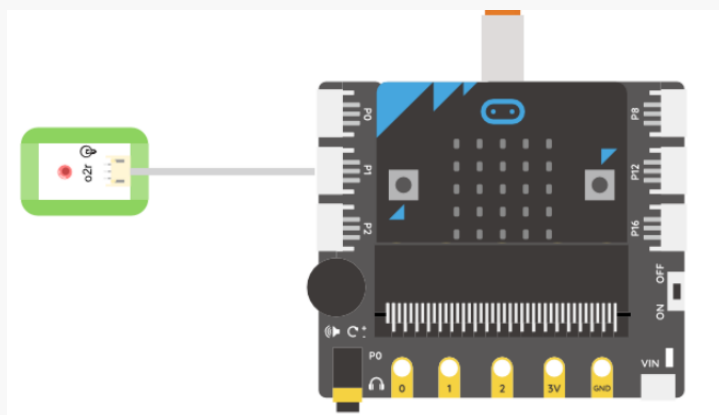
可移动门铃基于 micro:bit 的 Redio 功能，需要两块 micro:bit，一块是发射端，另外一块是可以移动的接收端。



发射端硬件连接：



接收端硬件连接：

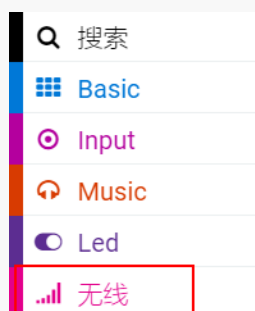


编写程序

任务一：设置发射端程序

STEP1：基于已有的编程基础，我们已经掌握对于按钮与运动传感器这两种输入模块的编程方法。可移动门铃的发射端需要将是否门外有人的消息，远程发射出去，怎么实现呢？这就是这个项目中需要重点学习的知识：无线传输。

在指令区中有“无线”指令模块。

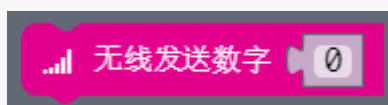


无线功能的使用中有三个常见的模块：

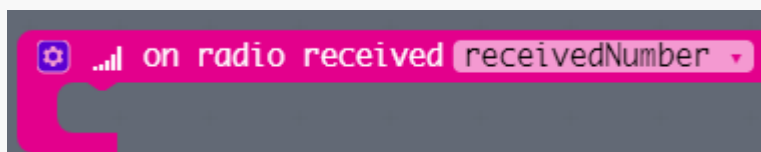
设定信号组：



无线发射信号：



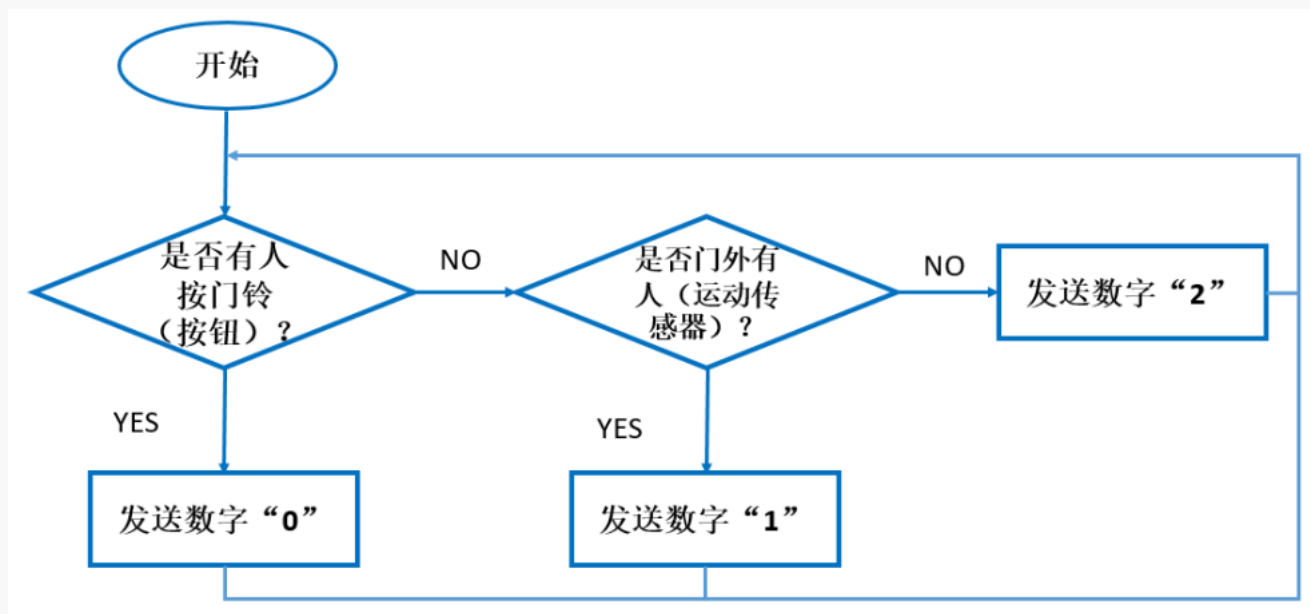
无线接收信号：



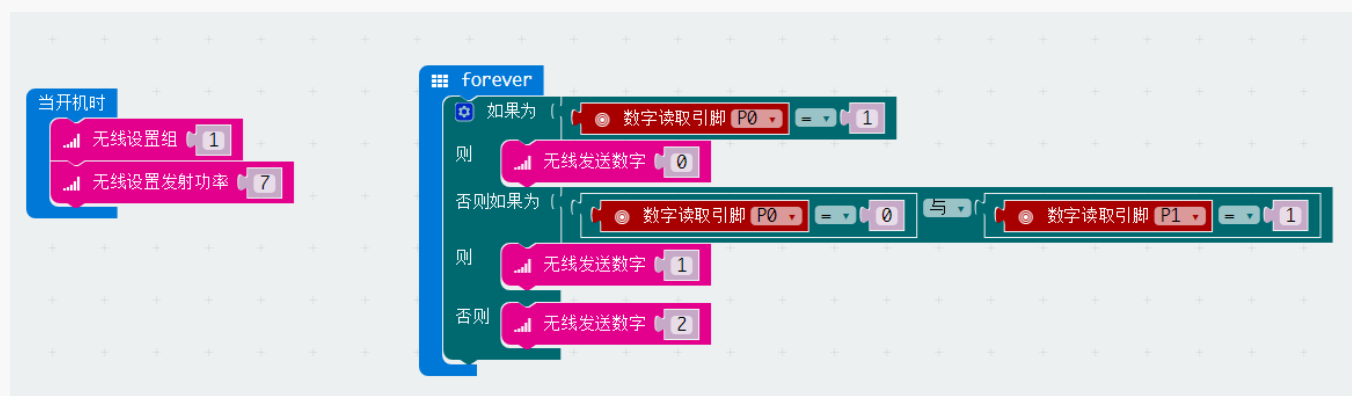
接收端需要识别接收到的信号，做出对应的动作，这里对应发射端有数字和文字两种情况。

注意，“receivedNumber”代表接收到的当前数字。

STEP2: 了解了无线功能的运用方法，对发射端进行程序编辑。编程流程图。

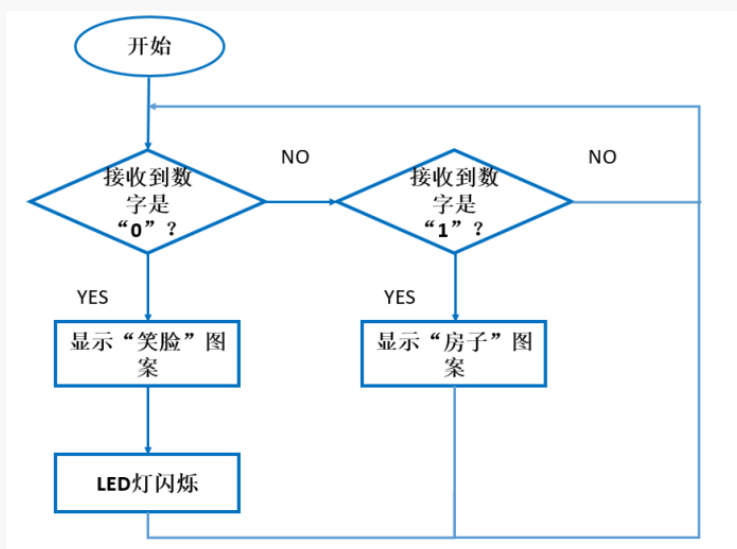


STEP3: 发射端完整程序。

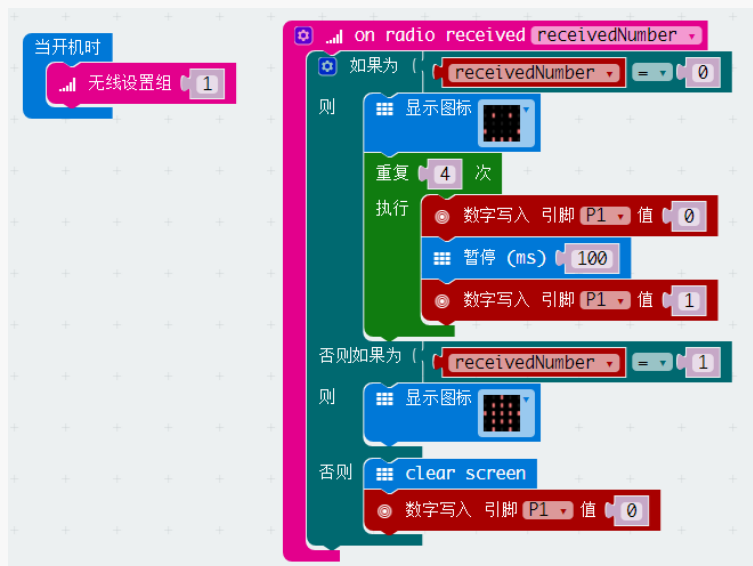


任务二：设置接收端程序

STEP1: 接收端功能流程图。



STEP2: 接收端完整程序。



能力提升

在门外按门铃的人一定很焦急，请为门外的门铃，设定“请稍等”的声音。

项目四 密室逃脱

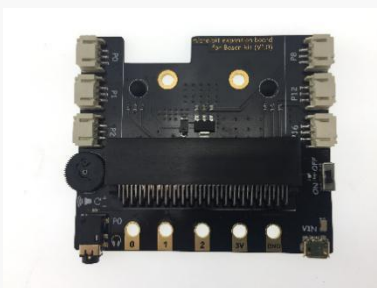
密室逃脱游戏，听起来是不是很有趣呢？那么你有玩过密室逃脱游戏吗？没有玩过也不要伤心，因为我们接下来就可以自己制作一个密室逃脱游戏的模拟装置啦！

所需元件

1 × micro:bit 主控板



1 × micro:bit 扩展板



1 × 彩虹灯带

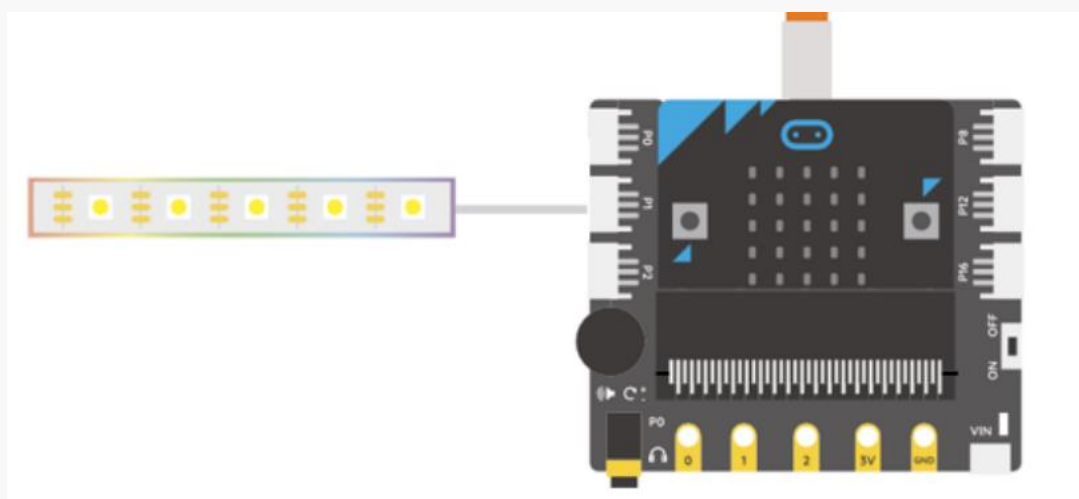


1 × USB A to B 连接线



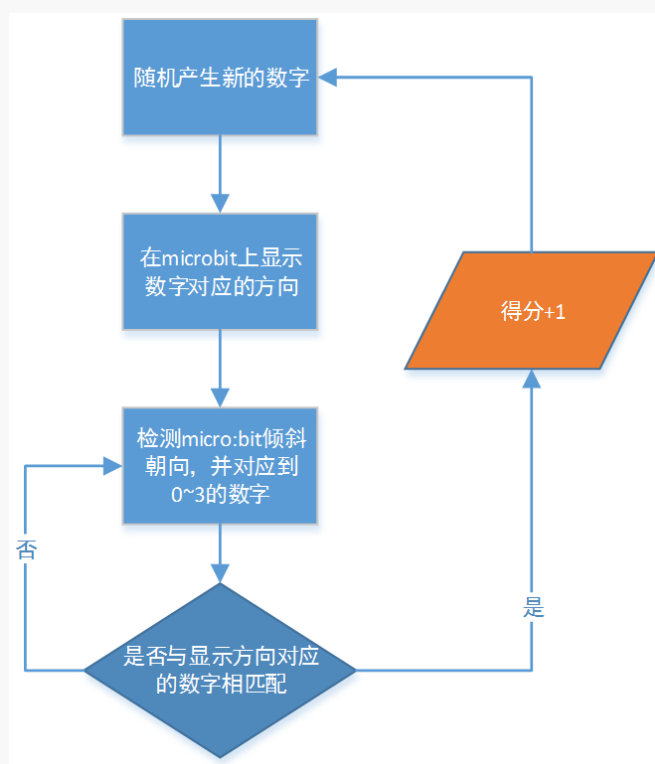
硬件连接

将彩虹灯带接在扩展板的 P1 号接口。



编写程序

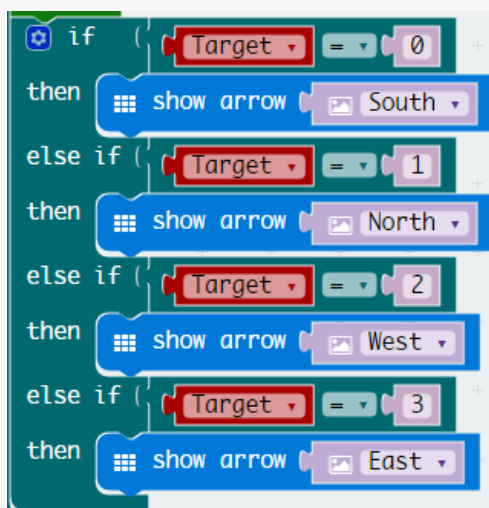
STEP1: 首先我们需要让 micro:bit 能够随机显示方向，并在我们向该方向倾斜时能识别出来，随后随机显示下一个方向。这样的程序的流程图大致如下。



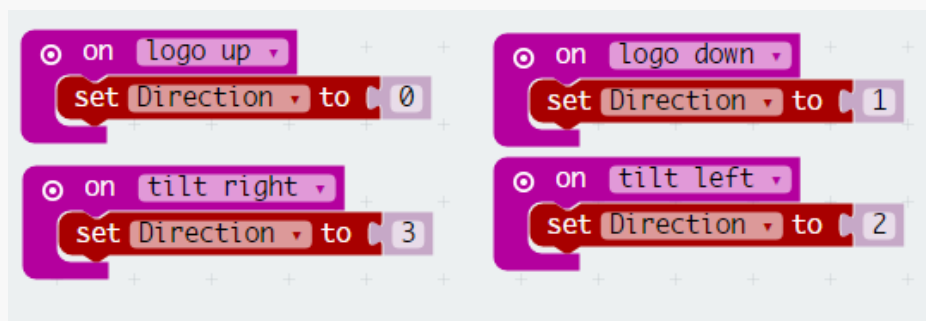
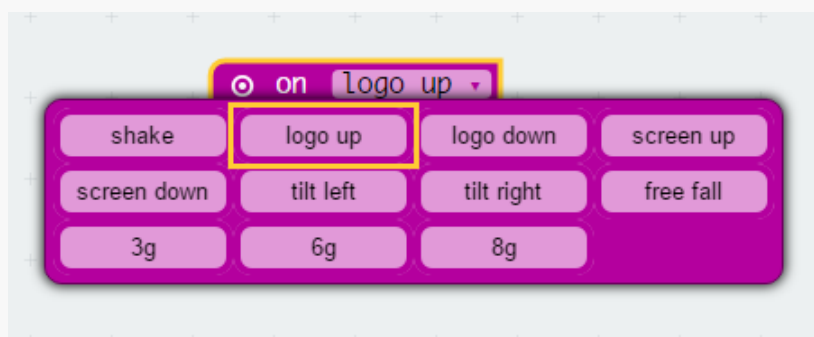
可以用 0~3 四个数字代表四个方向，使用 Math 下的子模块 “Pick Random” 来从 0~3 四个数字中随机选择一个数字。



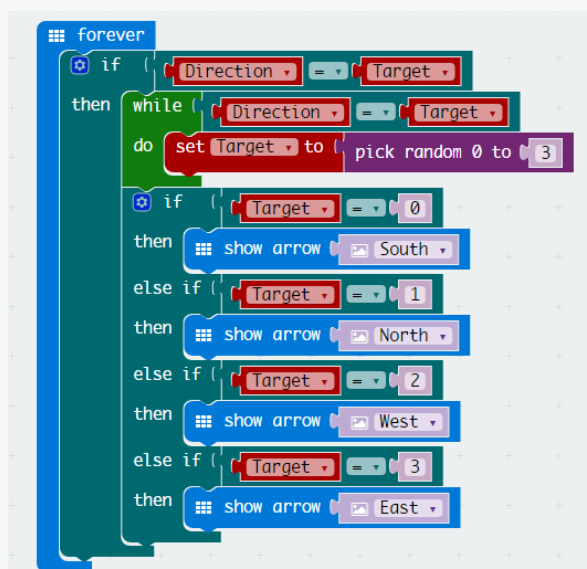
STEP2: 随后通过 if 语句将对应的方向显示在屏幕上。



STEP3: 通过在 Input 下的姿势识别模块来读取 micro:bit 的倾斜方向，并且分别对应到 0~3 中。

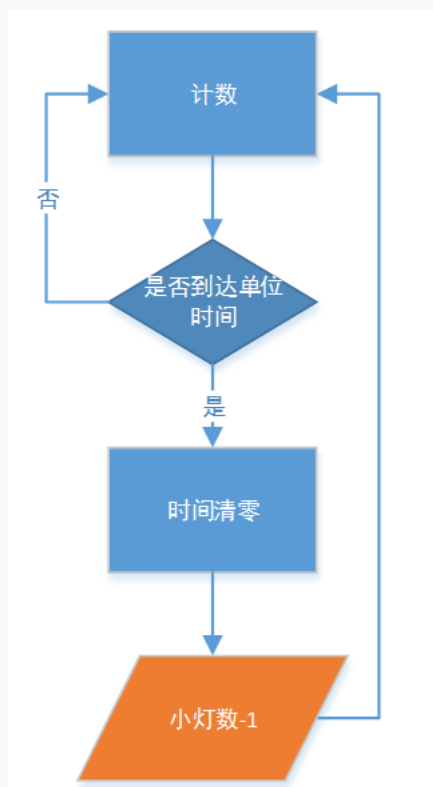


STEP4: 通过一个 if 语句判断是检测到的方向是否正确。



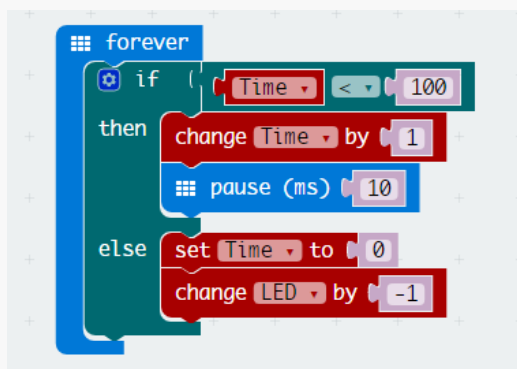
注意：While 语句用于保证两次随机出现的方向不相同。

STEP5：接下来我们需要一个倒计时的功能，让导火线持续燃烧。倒计时的流程大致如下。

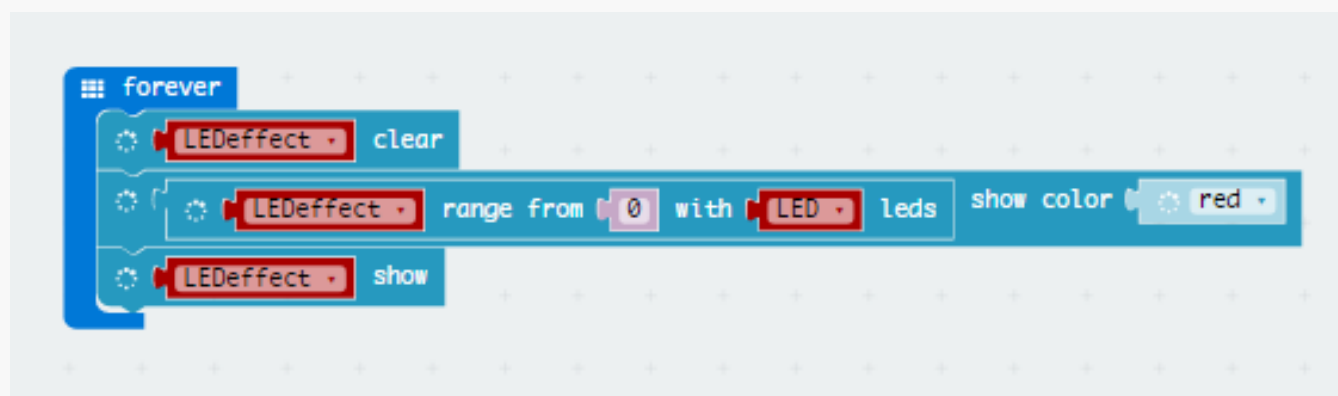


可以用以下模块组合来实现该功能。

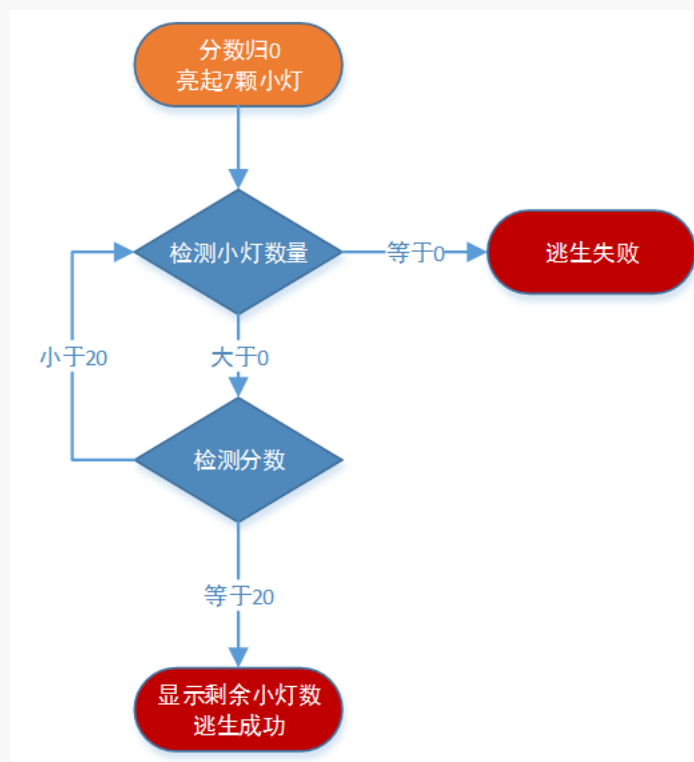
注意：因为 micro:bit 在处理数据时会消耗到一定的时间，所以倒计时模块实际运行时走得慢一些。



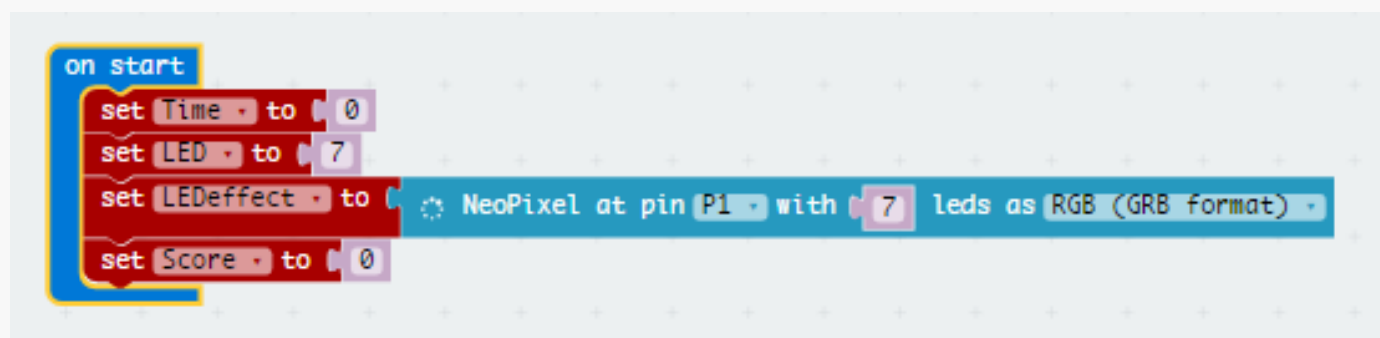
小灯的显示部分：



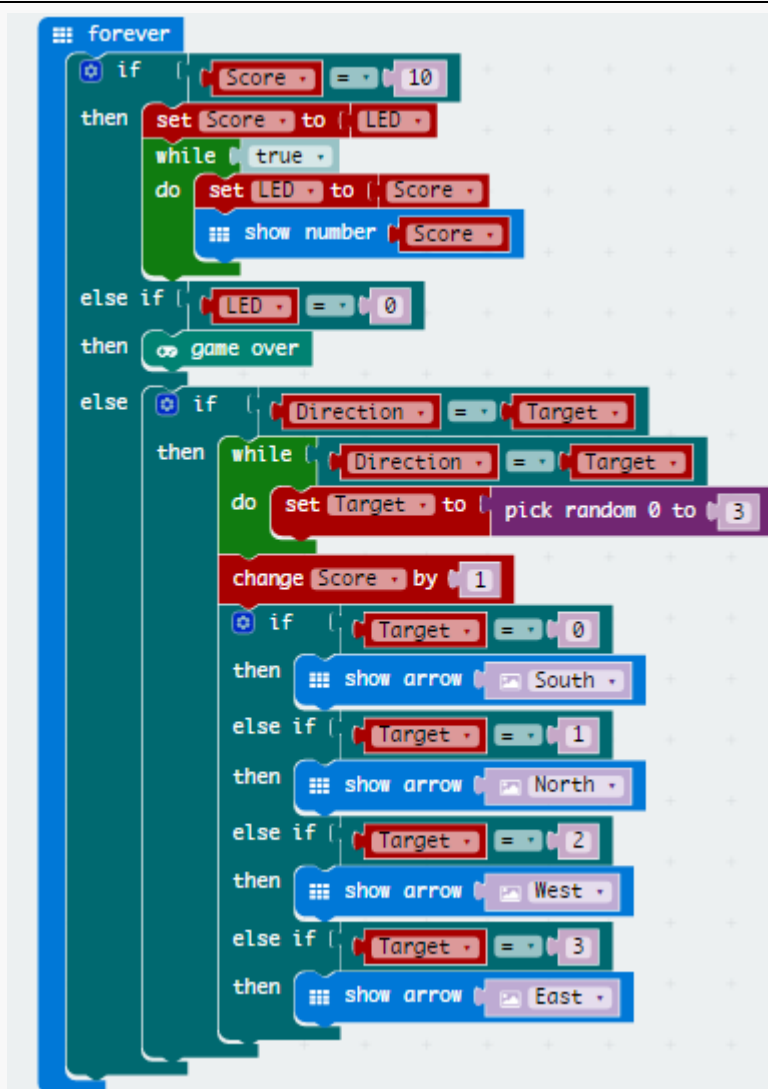
STEP6: 我们还需要增加计分功能。流程图大致如下。



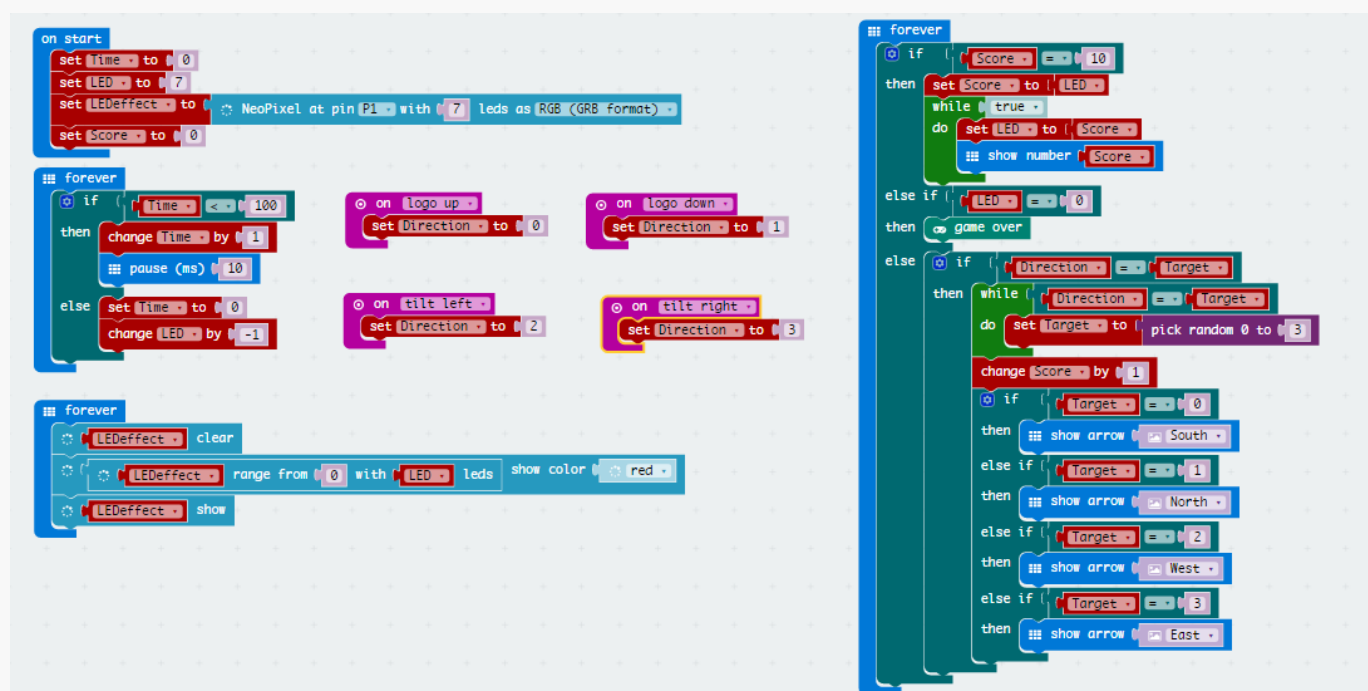
清零部分的程序：



STEP7: 计分部分可以合并在主程序中。



STEP8: 最终程序。



能力提升

尝试对 LED 灯效进行编程，在剩余时间越少的时候，闪烁得越快；

尝试加快 LED 等的熄灭速度，增加游戏难度；

将上下左右方向用其他图形替代，挑战更高难度的下的反应力。

希望你的 **Micro:bit** 之旅不会因此而停止，用你的奇思妙想，玩出更多新颖有创意的作品。如果你愿意与我们分享的话，也可以直接登陆我们的论坛，让我们的社区论坛记录下你的点点滴滴！

欢迎登陆 DFRobot 创客社区！

DFRobot 创客社区: www.dfrobot.com.cn